ISSN 0868-6157

КОМПЬЮТЕР ПРЕСС



ОДИН ИЗ ВЕДУЩИХ
ОТЕЧЕСТВЕННЫХ ПРОИЗВОДИТЕЛЕЙ
СПЕЦИАЛИЗИРОВАННОЙ МЕБЕЛИ
ДЛЯ КОМПЬЮТЕРНЫХ ЦЕНТРОВ,
ОФИСОВ, БАНКОВ И БИРЖ



ДОК-17

НА ВАШЕМ РАБОЧЕМ МЕСТЕ
ВСЕ БУДЕТ ПОД РУКОЙ,
ЕСЛИ ВЫ ИСПОЛЬЗУЕТЕ
КОМПЛЕКТЫ «КЛЕРК»
И «ОФИС-МОДУЛЬ»

129128 Москва, ул. Ростокинская, 2. Телефон: (095) 191-06-43, 181-04-67
Факс: (095) 187-98-62 E-mail: dok@aquillon.msk.su

Совместное советско-американское предприятие «СОВАМИНКО»

КОМПЬЮТЕР ПРЕСС

8'92



КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
8'92 (32)

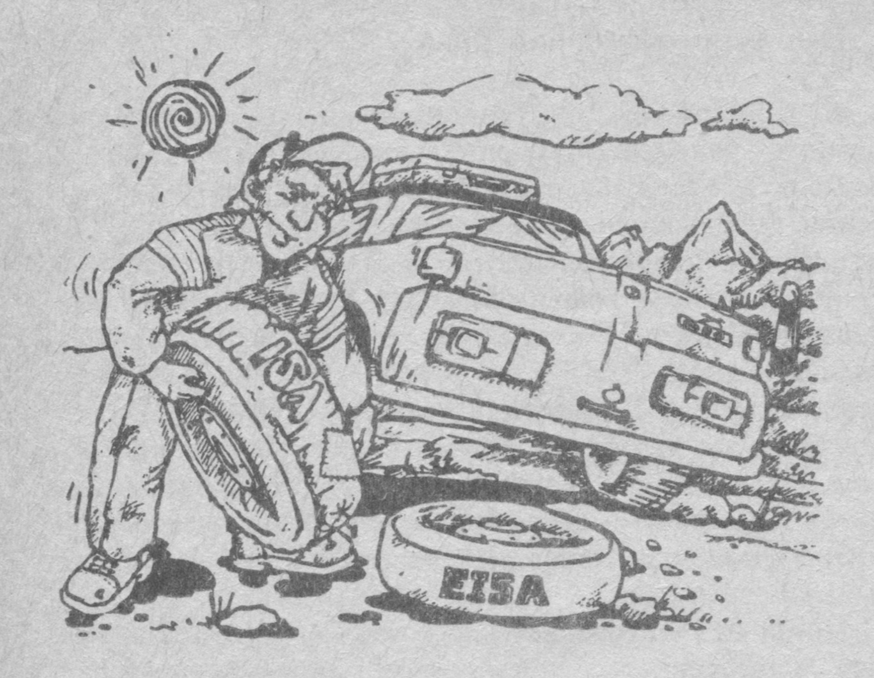
Адрес редакции:
113093 Москва, аб.ящик 37
Факс: (095) 200-22-89
Телефон для справок: (095) 471-32-63
E-mail: editorial@computerpress.msk.su

Сдано в набор 22.06.92. Подписано к печати 10.07.92.
Формат 84x108/16. Печать офсетная. Бумага типографская. Усл.печ.листов 8,4+0,42 (обл.).
Тираж 62000 экз.
Заказ 2952. С-8.

Оригинал-макет подготовлен агентством «КомпьютерПресс».

Журнал сверстан на оборудовании фирмы Summit Systems.

Отпечатано в полиграфической фирме «Красный пролетарий» РГИИЦ «Республика».
103473 Москва, И-473, Краснопролетарская, 16.



В 1989 году на осенней выставке COMDEX в Лас-Вегасе был представлен новый стандарт EISA — стандарт для 32-разрядной системной шины, альтернативной "микроканалу" MCA фирмы IBM. Для каких же технических приложений целесообразно использование этой шины и какие преимущества она имеет по сравнению со "старой доброй" шиной ISA?

Шина EISA

Специалистам, так или иначе имевшим дело с большими ЭВМ, известно, что для операций ввода-вывода в этих компьютерах используются обычно специальные периферийные процессоры. Освобождая главный (host) процессор ЭВМ от многих рутинных операций, не связанных с основными решаемыми задачами, эти процессоры позволяют существенно повысить производительность вычислительной системы в целом. Архитектура персональных ЭВМ, как правило, не предусматривает использования дополнительных процессоров — все операции ввода-вывода происходят через регистры основного процессора. Этим отчасти объясняются невысокие производительность и стоимость "персоналок". Коротко напомним, о чем идет речь.

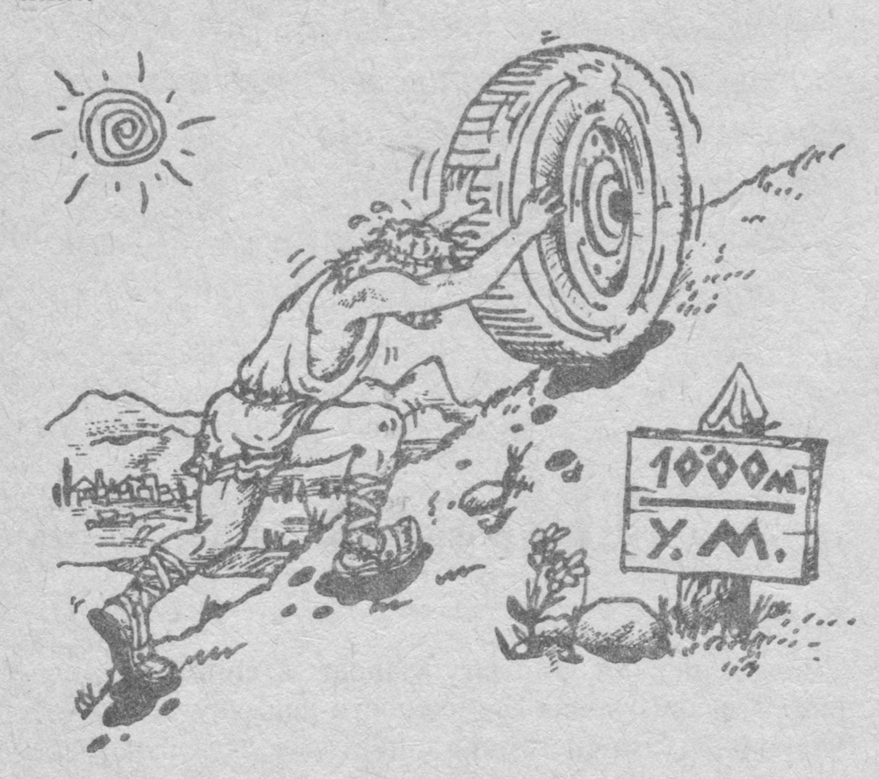
Шаги эволюции

Системная шина IBM PC и IBM PC/XT была предназначена для одновременной передачи только 8 разрядов данных, так как используемый в компьютерах микропроцессор i8088 имел 8 линий данных. Кроме этого, системная шина включала 20 адресных линий, которые ограничивали адресное пространство пределом в 1 Мбайт. Для работы с внешними устройствами в этой шине были предусмотрены также 4 линии аппаратных прерываний (IRQ) и 4 линии для требования внешними устройствами прямого доступа в память (DMA, Direct Memory Access). Скорость передачи данных с использованием 8-разрядного DMA составляла от 250 000 до 500 000 бит/с.

В компьютерах PC/AT, использующих микропроцессор i80286, впервые стала применяться новая системная шина ISA (Industry Standart Architecture), полностью реализующая возможности упомянутого микропроцессора. Эта системная шина позволяла передавать параллельно уже 16 разрядов данных, а благодаря 24 адресным линиям напрямую обращаться к 16 Мбайтам системной памяти. Количество линий аппаратных прерываний в этой шине было увеличено с 7 до 15, а каналов DMA — с 4 до 7. Надо отметить, что новая системная шина ISA полностью включала в себя возможности старой 8-разрядной шины. Скорость пересылки данных с использованием DMA для 8-разрядных режимов составляла от 100 000 до 400 000 бит/с, а для 16-разрядных — от 400 000 до 800 000 бит/с. Некоторое ухудшение характеристик передачи данных при использовании 8-разрядного режима было связано с необходимостью обеспечения совместимости с платами "старых" 8-разрядных адаптеров, поскольку все такие устройства, используемые в PC/XT, могли без проблем применяться и в PC/AT 286.

Однако с появлением новых микропроцессоров, таких как i80386 и i80486, стало очевидно, что одним из вполне преодолимых препятствий на пути повышения производительности компьютеров с этими микропроцессорами является системная шина ISA. Дело в том, что возможности ISA-шины для построения высокопроизводительных систем следующего поколения были практически исчерпаны. Новая системная шина должна была обеспечить наибольший возможный объем адресуемой памяти, 32-разрядную передачу данных, в том числе и в режиме DMA, улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расширения. Фирма IBM смогла технически реализовать эти и некоторые

другие требования в своей новой системной шине MCA (Micro Channel Architecture), которая, увы, была полностью несовместима с ISA. Пользователям нового ряда компьютеров PS/2, применяющих шину MCA, пришлось расстаться со всеми платами расширения из своих "старых" AT-ишек. Но об IBM — aut bene, aut nihil.



"Коллективная" системная шина

Консорциум из 9 фирм ("Команда Девяти", Gang of Nine), включающий в себя наиболее крупных производителей клонов IBM PC (AST, Compaq, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wyse и Zenith), предложил эволюционный подход для развития шины ISA — шину EISA (Extended Industry Standart Architecture). Таким образом, одной из важных предпосылок создания EISA было развитие такой 32-разрядной архитектуры, которая, в отличие от MCA, была бы не только совместима "сверху вниз" с шиной ISA, но и смогла бы добавить новые функциональные особенности, присущие высокопроизводительной шине. Кстати, сам проект стандарта EISA был получен от независимой американской фирмы BCPR Services. Надо отметить, что консорциум, предложивший стандарт EISA, ориентировался на вполне конкретную область применения новой архитектуры. А именно, на компьютеры, оснащенные высокоскоростными подсистемами внешней памяти на жестких магнитных дисках с буферными кэш-ЗУ. Такие компьютеры используются в основном в качестве файл-серверов. Как правило, серверы с архитектурой EISA предоставляют пользователям вычислительную мощность на уровне миниЭВМ. В настоящее время известно уже большое количество подобных EISA-систем, предлагаемых различными фирмами. Из достаточно известных можно назвать, например, компьютеры Vectra 486 фирмы Hewlett-Packard, SystemPro фирмы Compaq, PowerVEISA фирмы ALR и т.д.

Как устроена шина EISA

В EISA-разъем на системной плате компьютера, помимо, разумеется, специальных EISA-плат, может вставляться либо 8-, либо 16-разрядная плата расширения, предназначенная для обыкновенной PC/AT. Это обеспечивается поистине гениальным, но простым конструктивным решением. EISA-разъемы имеют два ряда контактов, один из которых (верхний) использует сигналы шины ISA, а второй (нижний) — соответственно, EISA. Контакты в соединителях EISA расположены так, что рядом с каждым сигнальным контактом находится контакт "земля". Благодаря этому сводится к минимуму вероятность генерации электромагнитных помех, а также уменьшается восприимчивость к таким помехам.

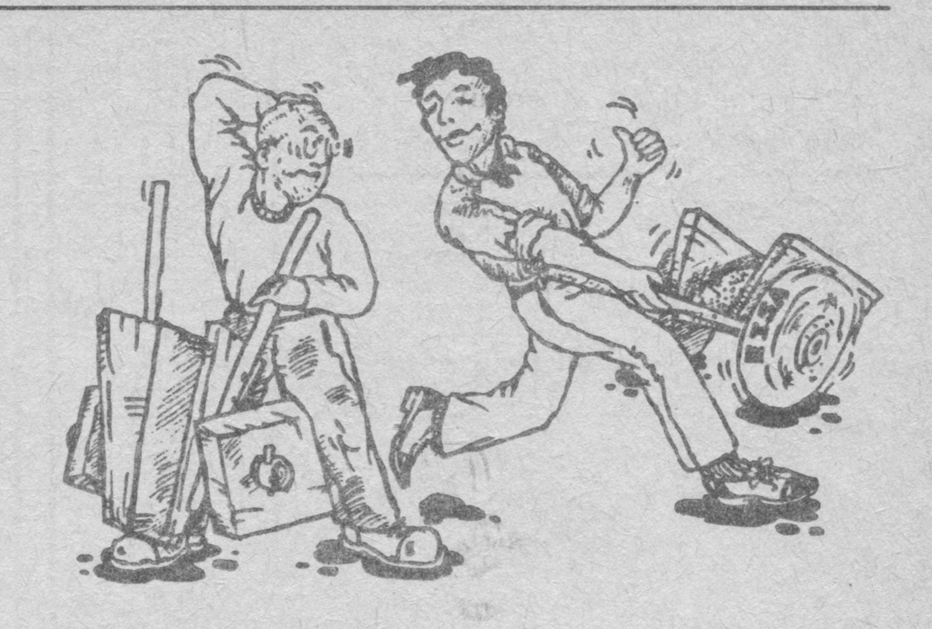
Шина EISA позволяет адресовать 4-Гбайтное адресное пространство, доступное микропроцессорам i80386/486. Однако доступ к этому пространству могут иметь не только центральный процессор, но и платы управляющих устройств типа bus master (то есть устройства, способные управлять передачей данных по шине), а также устройства, имеющие возможность организовать режим DMA. Стандарт EISA поддерживает многопроцессорную архитектуру для "интеллектуальных" устройств (плат), оснащенных собственными микропроцессорами. Поэтому данные, например, от контроллеров жестких дисков, графических контроллеров и контроллеров сети могут обрабатываться независимо, не загружая при этом основной процессор. Максимальная скорость передачи по шине в так называемом пакетном режиме (burst mode) может достигать 33 Мбайт/с. Не следует, конечно, путать этот режим с burst bus в микропроцессоре i80486. В обычном (стандартном) режиме скорость передачи по EISA-шине не превосходит, разумеется, известных значений для ISA-шины.

На шине EISA предусматривается метод централизованного управления, организованный через специальное устройство — системный арбитр. Таким образом поддерживается использование ведущих устройств на шине, однако предусматривается также предоставление шины запрашивающим устройствам по циклическому принципу. В соответствии с этим шина поочередно или по мере надобности предоставляется для регенерации динамической памяти (refresh DRAM), а также устройствам, требующим DMA, устройствам bus master и основному процессору.

В стандарте EISA предусматривается не более 15 мест для подключения плат расширения. От каждого такого места имеется специальная линия к системному арбитру, через которую устройство может потребовать доступ к шине. Для каждой такой линии, обозначаемой MREQx (где x определяется по номеру слота), имеется соответствующая линия ответа MAKx. Управляющее устройство (bus master) может активизировать сигнал MREQx не более 64 тактов шины, что соответствует приблизительно 8 мкс, после чего происходит освобождение шины. В противном случае на шине

возможна ситуация тайм-аута (timeout), которую можно прервать только подачей сигнала начальной установки RESET. Таким образом, для плат, выполняющих функции устройств типа bus master, устанавливается предельное время ожидания.

На шине EISA основной процессор или устройство bus master могут устанавливать сигнал Lock. Активизация этого сигнала возможна, однако, в течение не более 8 мкс. В это время активное устройство имеет исключительные права на шине. Такое свойство особенно важно для систем реального времени, которые используют для синхронизации процессов так называемые "семафоры". Установка сигнала Lock препятствует перезаписи одного и того же "семафора" разными устройствами, поскольку это, вообще говоря, может привести к краху системы.



Использование DMA и прерываний

Как и для шины ISA, в системе EISA имеется 7 каналов DMA. Выполнение DMA-функций полностью совместимо с аналогичными операциями на ISA-шине, хотя они могут происходить и несколько быстрее. Контроллеры DMA имеют возможность поддерживать 8-, 16- и 32-разрядные режимы передачи данных. В общем случае возможно выполнение одного из четырех циклов обмена между устройством DMA и памятью системы. Это — ISA-совместимые циклы, использующие для передачи данных 8 тактов шины; циклы типа A, исполняемые за 6 тактов шины; циклы типа B, выполняемые за 4 такта шины, и циклы типа C (или burst DMA), в которых передача данных происходит за один такт шины. Типы циклов A, B и C поддерживаются 8-, 16- и 32-разрядными устройствами, причем возможно автоматическое изменение размера (ширины) данных при передаче в не соответствующую размеру память. Большинство ISA-совместимых устройств, использующих DMA, могут работать почти в 2 раза быстрее, если они будут запрограммированы на применение циклов A или B, а не стандартных (и сравнительно медленных) ISA-циклов. Такая производительность достигается только путем улучшения арбитрирования шины, а не в ущерб совместимости с ISA. Приоритеты DMA в системе могут быть либо "вращающимися" (переменными), либо жестко установленными. DMA-функции предоставляются обычно для плат, которые имеют средние показатели по производительности и не оснащены собственными микропроцессорами. По сути, такие платы представляют приемлемую по стоимости альтернативу платам, обладающим возможностями bus master.

Линии прерывания шины ISA, по которым запросы прерывания передаются в виде перепадов уровней напряжения (фронтов сигналов), сильно подвержены импульсным помехам. Поэтому в дополнение к привычным сигналам прерываний на шине ISA, активным только по своему фронту, в системе EISA предусмотрены также сигналы прерываний, активные по уровню. Причем для каждого прерывания выбор той или иной схемы активности может быть запрограммирован заранее. Собственно прерывания, активные по фронту, сохранены в EISA только для совместимости со "старыми" адаптерами PC/AT, обслуживание запросов на прерывание которых производит схема, чувствительная к фронту сигнала. Понятно, что прерывания, активные по уровню, менее подвержены шумам и помехам, нежели обычные. К тому же (теоретически), по одной и той же физической линии можно передавать бесконечно большое число уровней прерывания. Таким образом, одна линия прерывания может использоваться для нескольких запросов. Запрашивающее прерывание устройство выставляет на соответствующей линии определенный уровень напряжения и удерживает его до тех пор, пока не будет обслужено.

Включение новых устройств

Каждому пользователю стандартной PC/AT известно, что включение нового адаптера устройства в состав компьютера связано с определенными трудностями, так как для этого, как правило, необходимо правильно установить адреса ввода-вывода, номера прерываний, DMA и т.п. Для компьютеров с шиной EISA предусмотрено автоматическое конфигурирование системы, которое происходит следующим образом. Каждый изготовитель плат расширения для компьютеров с шиной EISA поставляет вместе с этими платами и специальные файлы конфигурации. Информация из этих файлов используется на этапе подготовки системы к работе, которая заключается в разделении ресурсов компьютера между отдельными платами. Для каждой платы расширения EISA предусмотрен свой идентификационный номер. Понятно, что для "старых" плат адаптеров пользователь должен сам подобрать правильное положение DIP-переключателей и перемычек, однако сервисная программа на EISA-компьютерах позволяет отображать установленные положения соответствующих переключателей на экране монитора и дает некоторые рекомендации по правиль-

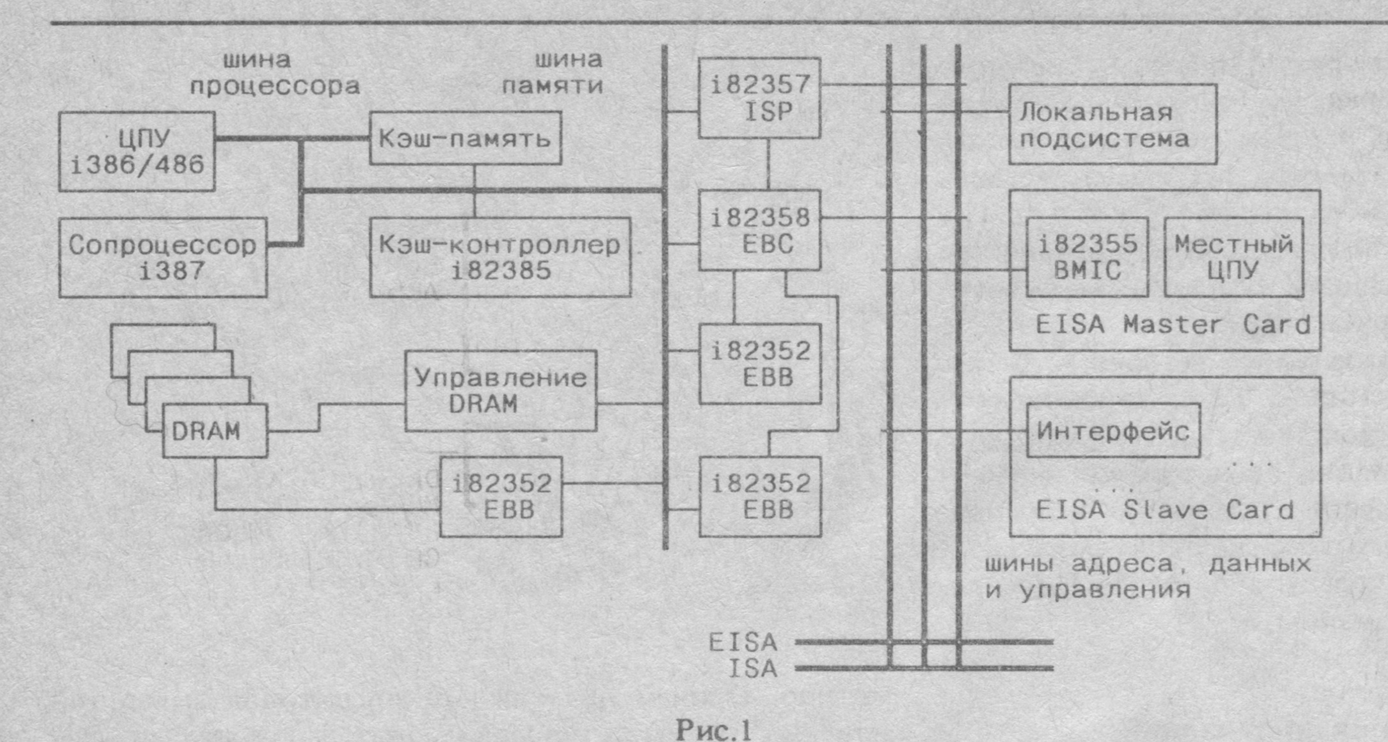


Рис.1

Контроллер шины EBC

Контроллер шины EBC действует практически как мост между платами ISA- и EISA-адаптеров на комбинированной ISA/EISA-шине. Примерная блок-схема этого контроллера приведена на рис. 2. Он управляет циклами передачи между EISA-, ISA- и CPU-шинами, а также следит за выполнением циклов на ISA/EISA-шине. Если основной процессор начинает цикл шины, а управляемое устройство не отвечает, то контроллер EBC передает управление другому устройству, подключенному к ISA/EISA-шине. Кроме того, этот контроллер генерирует управляющие сигналы для буферов адреса и данных, которые расположены между двумя этими шинами. Если необходимо, контроллер EBC может вставлять временные задержки между следующими друг за другом циклами ввода-вывода, инициированными основным процессором. Трансляцию циклов между ISA-, EISA- и основной шиной также поддерживает контроллер EBC. Кроме того, он преобразует циклы шины от устройств, управляющих EISA-шиной в данный момент. Таким устройством может быть как основной процессор, так и устройство bus master. В случае, если размер данных управляемого устройства ввода-вывода и основной памяти не совпадает, контроллер EBC выполняет "сборку" или "разборку" данных по байтам. Такая ситуация может иметь место, например, тогда, когда из основной

ной их установке. Помимо этого в архитектуре EISA предусматривается выделение определенных групп адресов ввода-вывода для конкретных слотов шины, что также позволяет избежать конфликтов между отдельными платами EISA.

Набор микросхем для EISA

Понятно, что широкие возможности шины EISA без использования набора соответствующих микросхем — не более чем прекрасная мечта. Реализовать сложный интерфейс на стандартных, например, ТТЛ-микросхемах при заданных ограничениях практически невозможно. Фирма Intel, как всегда, занимает лидирующее положение в разработке и изготовлении самых изощренных "чипсов". В настоящее время известно два набора микросхем "от Intel", которые предназначены для создания компьютеров с шиной EISA, это — наборы микросхем 82380 и 82380DT. Упрощенная блок-схема системы EISA с использованием набора микросхем 82380 приведена на рис. 1.

Набор микросхем 82380 полностью поддерживает микропроцессоры i80386/486 с тактовой частотой 25 и 33 МГц, а также контроллер кэш-памяти i82385DX. Применение внешней кэш-памяти допускается при использовании как микропроцессора i80386, так и i80486. Системная плата с шиной EISA на процессоре i80386 включает три микросхемы из набора 82350:

- контроллер шины i82358 (EISA Bus Controller, EBC);
- периферийный контроллер i82357 (Integrated System Periphal Chip, ISP);
- шинный буфер i82352 (EISA Bus Buffer, EBB).

Для плат расширения, которые используют собственные микропроцессоры (bus muster), предусмотрена специальная микросхема контроллера i82355 — Bus Master Interface Controller (BMIC).

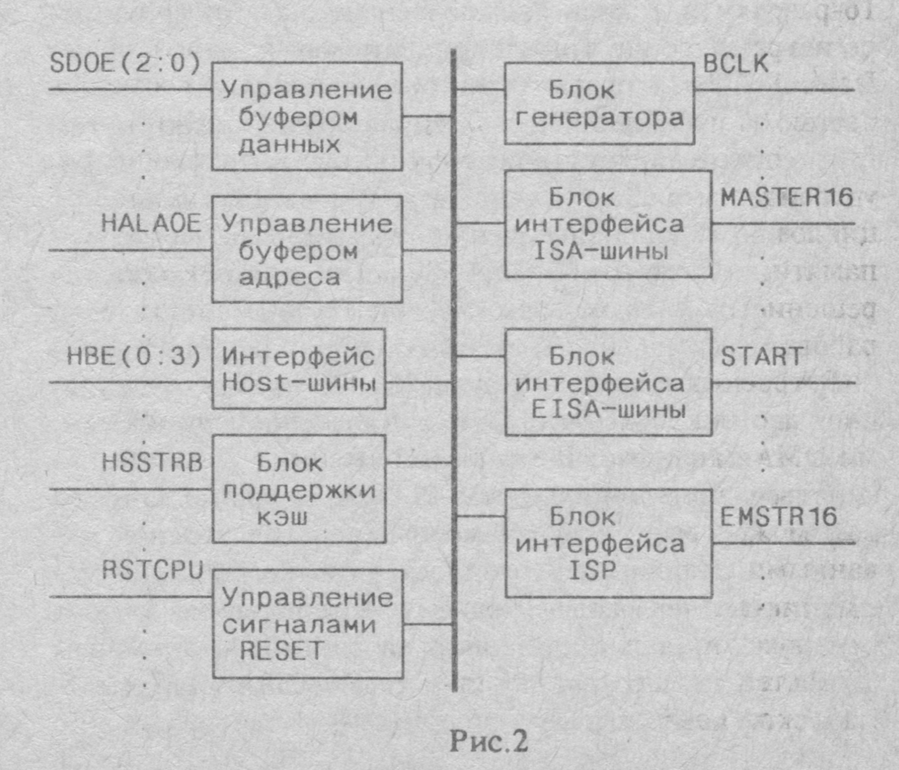


Рис.2

32-разрядной памяти надо переслать данные в 16-разрядное устройство. Логика обмена с шиной данных реализуется обычно на микросхеме i82352 (EBB), входящей в набор 82350.

Из синхросигнала процессора контроллер EBC генерирует тактовый сигнал для EISA-шины (BCLK), который может варьироваться в пределах от 6 до 8,33 МГц. Работа контроллера EBC тесно связана с функционированием периферийного контроллера ISP. От ISP к EBC, в частности, передаются требования запросов DMA и регенерации динамической памяти.

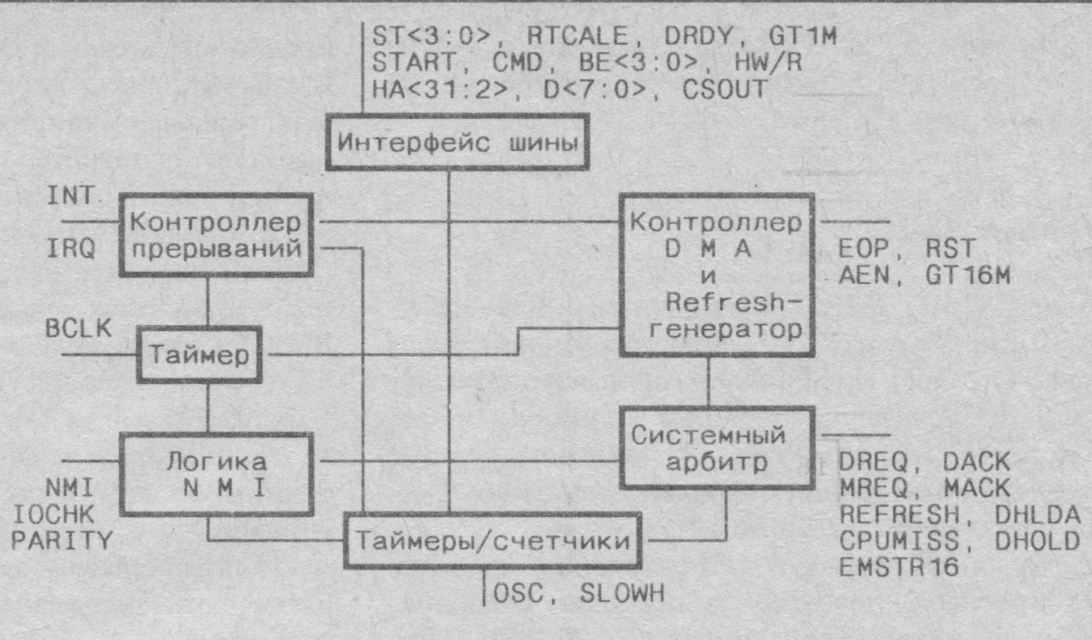


Рис.3

Периферийный контроллер ISP

Периферийный контроллер ISP, выполненный на микросхеме i82357, представляет собой многофункциональное устройство, которое объединяет различные функциональные блоки (рис. 3). Среди них 7-канальный контроллер DMA, 8-канальный контроллер прерываний и пять 16-битных программируемых счетчиков-таймеров. Кроме того, ISP служит в качестве центрального арбитра шины, к которому подключаются, в частности, все линии запросов MREQx и линии требования DMA. На шине EISA он может выступать и как управляющее (master), и как управляемое (slave) устройство. В режиме управляемого устройства все регистры контроллера могут читаться и во все эти регистры может производиться запись информации. Управляющее устройство на 16-разрядной ISA-шине может иметь доступ ко всем регистрам, совместимым с регистрами контроллера DMA i8237A, который, по сути, является составной частью контроллера ISP. В управляющем режиме ISP осуществляет все функции, присущие устройствам bus master, а именно: выполнение циклов DMA и циклов регенерации динамической памяти. Системный арбитр ISP принимает решение, в каком режиме будет происходить работа.

Встроенный контроллер DMA полностью заменяет две микросхемы i8237A с семью независимыми DMA-каналами. В его распоряжении находятся каналы с номерами 0-3 и 5-7. Каждый такой канал может быть запрограммирован для работы с данными шириной 8, 16 или 32 бита. Внутренний счетчик-таймер через каждые 15 мкс выдает соответствующий сигнал на контроллер DMA, который задействован для выполнения регенерации динамической памяти.

В ISP включены также два каскадно включенных контроллера прерываний, совместимых с i8259A. Пять счетчиков-таймеров, интегрированных в ISP, совместимы с известными микросхемами типа i8254.

Шинный буфер EBB

Шинный буфер EBB заменяет несколько микросхем типа 74F543/544, 74F657 и 74ALS245. Благодаря этому на системной плате компьютера, использующего микропроцессор i80386, можно сэкономить около 20 отдельных корпусов микросхем. В данном случае понадобится только три микросхемы i82352 (EBB).

Шинный буфер EBB может работать в одном из трех режимов.

- Режим 32-разрядных данных без контроля четности. В этом режиме шинный буфер EBB расположен между шиной данных основного процессора и шиной ISA/EISA. При этом возможна двунаправленная передача данных.
- Режим 32-разрядных данных с контролем четности. В этом случае EBB располагается на пути данных контроллера динамической памяти.

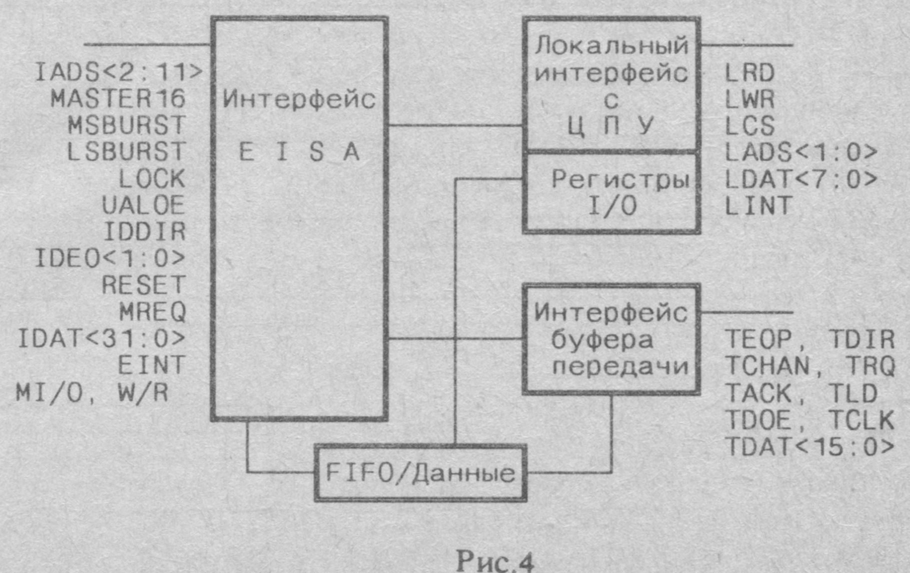


Рис.4

- Адресный режим EISA. EBB располагается на пути передачи адреса между шиной основного процессора и ISA/EISA-шиной.

Контроллер BMIC

Контроллер BMIC i82355 предназначен для EISA-плат, которые могут выполнять функции устройств bus master (рис. 4). Этот контроллер организует интерфейс между EISA-шиной, буфером передачи и локальным процессором на плате. В пакетном режиме (burst mode) этот интерфейс позволяет достигать скорости передачи данных до 33 Мбайт/с.

Интерфейс для передачи 16-разрядных данных использует протокол, похожий на протокол DMA, работающий с максимальной частотой 20 МГц. Его можно использовать оптимальным образом при применении быстрой статической памяти (SRAM). Этот интерфейс работает полностью асинхронно по отношению к шине EISA. Обмен данными между локальной памятью SRAM и шиной EISA может происходить только тогда, когда контроллер BMIC находится в режиме управляющего устройства.

В качестве локального процессора, осуществляющего, в частности, инициализацию и программирование BMIC, можно использовать, например, микропроцессор i80C186. По существу, интерфейс с процессором состоит из двух адресных линий и вывода прерывания. Используя адресацию через индексный регистр, можно обращаться к двум внутренним блокам регистров. Один из этих блоков будет доступен только локальному процессору, а второй, общий, может быть использован и от процессора, и от EISA-шины. Благодаря так называемой возможности peek-pook локальный процессор может иметь доступ к отдельным адресам памяти и ввода-вывода на шине EISA. Локальный процессор может также способствовать контроллеру BMIC выполнять передачу данных в одном из выбранных режимов (например, пакетном) между системной памятью основного процессора и SRAM локального процессора. Контроллер BMIC работает при этом в режиме управляющего.

Если на плате расширения не используется локальный процессор, то программирование контроллера BMIC осуществляет основной процессор через AT-совместимую часть EISA-шины.

В подчиненном (управляемом) режиме BMIC следит за обращениями к данной плате, в частности, к внутренним блокам регистров и портам ввода-вывода. В управляющем режиме контроллер BMIC может осуществлять передачу данных соответственно в пакетном и стандартном режимах, а также осуществлять режим peek-pook.

Режимы передачи данных

Управляющее устройство (bus master) на шине EISA может поддерживать два режима передачи данных между самим устройством и основной памятью или управляемым устройством ввода-вывода. Один из этих режимов называется стандартным, а другой — пакетным (burst). Для центрального процессора, вообще говоря, возможен еще один режим передачи — так называемый сжатый (compressed). В стандартном режиме работа устройства bus master полностью синхронизирована тактовым сигналом на системной шине BCLK. Частота этого сигнала обычно лежит в пределах от 6 до 8,33 МГц.

Стандартный режим

Когда устройство bus master приступает к передаче данных, оно запрашивает у системного арбитра разрешение на управление шиной, активизируя для этого линию MREQx (рис. 5). После этого bus master начинает опрашивать линию MAKx, посредством которой арбитр сообщает ему о разрешении на захват шины. Адрес, устанавливаемый на соответствующих линиях, указывает на объект передачи, которым может быть либо системная память, либо устройство ввода-вывода.

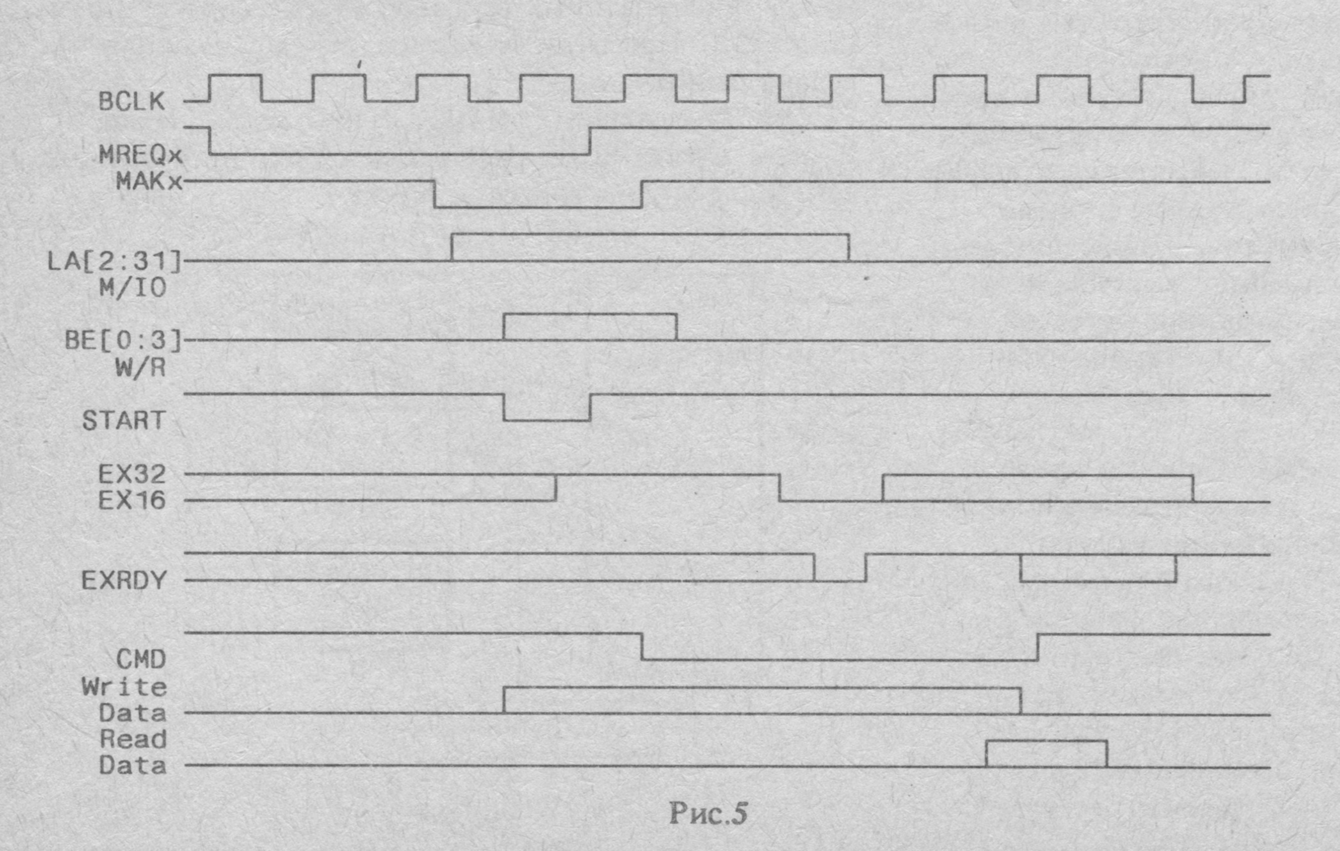


Рис.5



117607, Москва, Мичуринский пр-т, 31/1
тел. (095) 932-69-13
(095) 931-79-33

Предприятиям, Организациям и Фирмам

Пользователям системы КЛИППЕР

Корпорация Софткит

представляет

ТУЛКИТ для КЛИППЕРа

Инструментальный пакет программ для профессиональных пользователей системы КЛИППЕР

К Л И П К И Т

Корпорация Софткит Версия 1.0, 1992 г., Москва

Инструментальный пакет "**Клипкит**" Версия 1.0

Представляет собой мощное инструментальное средство для профессиональных программистов - разработчиков прикладных систем и АРМов на базе КЛИППЕР.

Пакет включает библиотеку программ с более чем 500 пользовательскими функциями и рядом сервисных утилит, которые обеспечивают функционально полный набор интерфейсных средств для КЛИППЕРа, существенно расширяющих и упрощающих использование его базовых возможностей, повышающих качество разрабатываемых прикладных систем.

Пакет включает следующие основные направления средств:

- ОКНА
- МЕНЮ
- СООБЩЕНИЯ
- УСТРОЙСТВА
- МЫШЬ, КУРСОР
- УПРАВЛЕНИЕ ПАМЯТЬЮ
- КЛАССЫ ОБЪЕКТОВ
- ГРАФИКУ
- КОММУНИКАЦИИ
- СЕРВИС

"В ХОРОШИХ ИНСТРУМЕНТАХ НУЖДАЮТСЯ ВСЕ, НО В ПЕРВУЮ ОЧЕРЕДЬ - ВЕЛИКИЕ МУЗЫКАНТЫ"

Страдивари

2 Зак.№2952

Далее управляющее устройство определяет режим передачи (запись или чтение) и размер передаваемых данных (одновременно могут передаваться не более 32 разрядов).

Начало цикла передачи bus master определяет по фронту сигнала START, который длится один период сигнала BCLK, после чего управляющее устройство инвертирует сигнал на линии CMD. Чтение сигналов EX16/EX32 позволяет управляющему устройству определить, является ли подчиненное устройство 16- или 32-разрядным. Активация сигнала EXRDY от управляемого устройства означает его готовность завершить цикл передачи. После этого bus master снова инвертирует значение сигнала CMD, обозначая этим завершение цикла передачи.

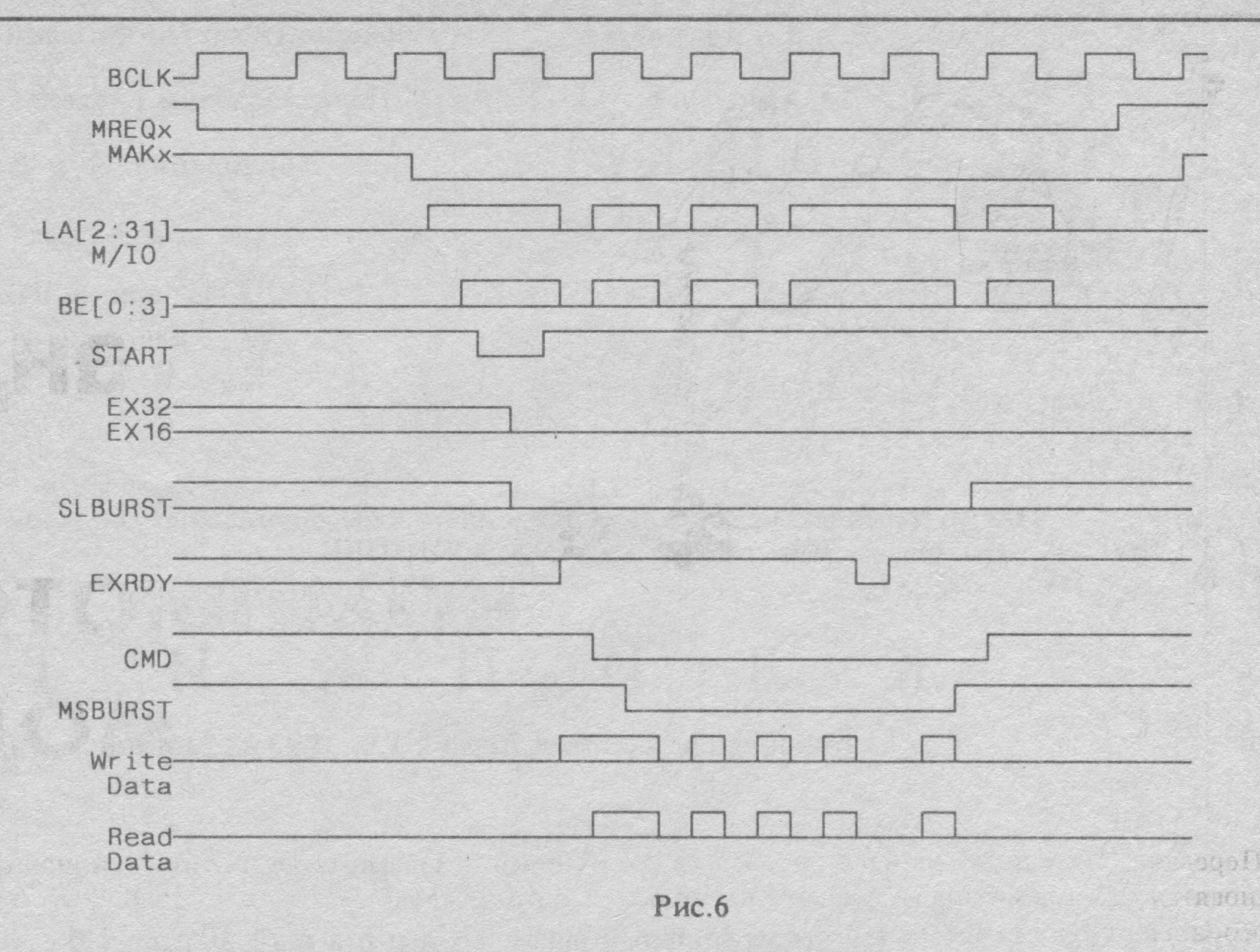


Рис.6

Во время записи данные на соответствующих линиях становятся действительными сразу после активизации сигнала START. При выполнении операции чтения управляющее устройство может читать данные по первому переднему фронту сигнала BCLK, после того, как стал активным сигнал EXRDY. Если подчиненное устройство не отвечает сразу, то bus master ждет следующего BCLK-цикла.

Пакетный режим

В то время как стандартный режим передачи требует, чтобы ведущее и ведомое устройства обменивались несколькими управляющими сигналами для записи или чтения каждого 16- или 32-разрядного числа, в пакетном режиме такой обмен происходит только один раз — в начале передачи. В пакетном режиме, помимо обычных управляющих сигналов, на EISA-шине используются специальные линии MSBURST и SLBURST (рис. 6).

Обмен сигналами при burst-цикле происходит в той же последовательности, что и при стандартном цикле передачи, но только до того момента, пока на шине не активизирован сигнал START. С этого момента bus master начинает опрашивать сигнал SLBURST от подчиненного устройства. Как только этот сигнал становится активным, управляющее устройство инвертирует сигнал на шине MSBURST и начинает именно burst-цикл. Данные, передаваемые ведущим устройством, являются действительными сразу после начала сигнала START. Данные, принимаемые от подчиненного устройства, действительны до окончания сигнала CMD. В течение всего burst-цикла необходимые уровни управляющих сигналов на EISA-шине должно обеспечивать устройство bus master. Цикл передачи заканчивается после того, как ведущее устройство инвертирует сигналы MSBURST и CMD.

В заключение хотелось бы сказать несколько слов о втором наборе микросхем для EISA фирмы Intel — 82350DT. Этот набор предназначен для компьютеров, имеющих модульную архитектуру, когда шина памяти (основная шина) работает асинхронно с тактирующим сигналом процессора. В набор 82350DT включены следующие новые микросхемы: контроллер шины EISA i82358DT, шинный буфер i82353, контроллер динамической памяти i82359 и локальный контроллер устройств ввода-вывода i82351.

А.Борзенко

В статье использованы материалы, предоставленные фирмой Intel



Стандарты для компьютерных модемов

Передача и прием файлов посредством модема становятся обычными процедурами для пользователей персональных компьютеров. В них нет ничего необычного, однако встречаются модемы, которые не могут обеспечить эффективную связь из-за проблем совместимости, так как не все они соответствуют одним и тем же стандартам.

У рядового пользователя обилие существующих модемных стандартов иногда вызывает проблемы. Посмотрев на рекламу модемов, вы увидите длинный перечень протоколов: Bell 103j, Bell 212a, V.22, V.22bis, V.32, V.42 и т.д. Данная статья поможет пользователю свободно ориентироваться во всем многообразии модемных стандартов, и надпись на коробке модема вроде

External Modem
Send & Resive FAX/Data
V.32/V.22bis/MNP5/V.42

не будет вызывать у вас паники.

Используемые сегодня стандарты для модемов созданы тремя организациями: Bell Standarts, CCITT Recommendations (или МККТТ) и EIA/TIA Standarts. В таблице перечислены стандарты для скоростей от 300 до 14400 бит/с, для работы в выделенных и коммутируемых телефонных линиях.

Таблица

Стандарт	Скорость	Линия	Дуплекс
Bell 103j	300	ком	полн
V.21	300	ком	полн
V.22	600	ком	полн
	1200	ком	полн
V.22bis	1200	ком	полн
	2400	ком	полн
V.23	600	ком	полу
	1200	ком	полу
Bell 212a	1200	ком	полн
Bell 202	1200	ком	полу
Bell 202	1200	выд	полн
V.26	2400	выд	полн
V.26bis	1200	ком	полу
	2400	ком	полу
V.26ter	1200	ком	полн
	2400	ком	полн
Bell 208	4800	выд	полн
V.27	4800	выд	полн
V.27bis	2400	выд	полн
	4800	выд	полн
V.27ter	2400	ком	полу
	4800	ком	полу
V.29	9600	выд	полн
V.32	9600	ком	полн
V.32bis	14400	ком	полн
V.33	14400	выд	полн

Низкоскоростные стандарты

Основными низкоскоростными стандартами являются Bell 103j на скорость 300 бит/с и Bell 212a на скорость 1200 бит/с. Почти каждый продаваемый модем поддерживает эти стандарты. Любой из данных стандартов может использоваться как для основной скорости, так и для пониженной. Пониженная скорость используется, если модем не в состоянии установить связь на основной, как правило, самой высокой скорости, обычно из-за слишком высокого уровня шумов в канале. Например, если модем пытается связаться на скорости 2400 бит/с, но определяет, что линия не сможет поддерживать такую скорость передачи, он может попытаться связаться на скорости 1200 или 300 бит/с.

Bell 103j и Bell 212a являются полностью дуплексными стандартами для 2-проводных линий связи. Это означает, что модемы, поддерживающие оба этих стандарта, одинаково используют линию. Они пере-

дают и принимают данные в обоих направлениях одновременно. Стандартной ширины телефонного канала (3000 Гц) как раз достаточно, чтобы данные передавались в обоих направлениях на скоростях до 1200 бит/с включительно.

Так как МККТТ (Международный Консультативный Комитет по Телефонии и Телеграфии) занимается развитием международных стандартов начиная с 1960 года (в то время как Bell был принят в качестве американского стандарта), то большинство каналов на 1200 бит/с используют стандарт, известный как V.22. Он подобен стандарту Bell 212a, но в нем используются различные частоты для передачи данных. Поэтому модемы V.22 и Bell 212a несовместимы между собой.

Для работы на скорости 2400 бит/с большинство компьютерных модемов пользуются рекомендацией V.22bis. Стандарт Bell для этой скорости никогда полностью не принимался. И как результат, сейчас существует целый ряд полностью совместимых 2400-битных модемов, базирующихся на V.22bis.

Как и более низкоскоростные стандарты, V.22bis — двухпроводный (то есть подключаемый к коммутируемой телефонной линии), дуплексный стандарт. Для приспособления двунаправленой передачи данных со скоростью 2400 бит/с в полосе 3000 Гц биты данных группируются по четыре и передаются одной посылкой. Каждая такая посылка передается со скоростью 600 бод, и два направления передачи удается совместить в одном телефонном канале.

Среднескоростные стандарты

До 1984 года передача данных со скоростями свыше 2400 бит/с была возможна только по выделенным (4-проводным) линиям. Специальные стандарты, такие как Bell 208 на 4800 бит/с, V.29 на 9600 и V.33 на 14400 бит/с, можно было использовать лишь на арендованных линиях. Однако оправдывать стоимость арендованных линий и дорогих модемов могли только пользователи, нуждающиеся в передаче большого количества данных.

В 1984 году МККТТ принял стандарт V.32 для использования в обычных коммутируемых телефонных линиях. V.32 скачкообразно изменяет скорость от 2400 до 9600 бит/с, что значительно увеличивает его пропускную способность по сравнению с V.22bis. Используя улучшенную технологию для обеспечения 9600-битной передачи по обычной телефонной линии, V.32 открывает активному пользователю двери в мир высокоскоростной связи и эффективного обмена данными с помощью модема.

Стандарт V.32

Требуемая для использования рекомендаций V.32 технология далась изготовителям модемов нелегко. Уровень технических разработок, необходимых для использования модемов V.32, по самым скромным подсчетам, в 100 раз выше, чем для V.22bis. Поэтому модемы V.32, полностью соответствовавшие стандарту, появились лишь через два года после его принятия.

Для передачи данных на скорости 9600 бит/с V.32 модемы группируют данные по 4 бита, и потом эти сгруппированные данные передаются со скоростью 2400 бод. Хотя имеется возможность для работы только одного 2400-бодного канала в телефонной линии с диапазоном 3000 Гц, рекомендация V.32 позволяет обоим модемам передавать данные одновременно по одной и той же линии. Для достижения качественного обмена на скорости 9600 бит/с в коммутируемой линии используется система компенсации эха. В каждом модеме V.32 есть специальные цепи для сравнения характеристик модема и линии. Так как условия в реальной телефонной сети постоянно меняются, это сравнение никогда не бывает идеальным. В результате передаваемый сигнал отражается от неоднородностей и возвращается обратно в модем. Для получения хорошего принимаемого сигнала с достаточным уровнем эти отражения убираются схемой компенсации эха еще до начала процесса обработки сигнала.

Поскольку за единицу времени модем V.32 передает больше информации, нежели низкосокоростной модем, его сигнал оказывается более слабым и его тяжелее детектировать. По этой причине V.32 использует улучшенную технику фазового матричного кодирования. Она позволяет модему проверять несколько последовательно принимаемых сигналов и сравнивать их с известными образцами перед принятием решения о значении сигнала. Этот эффект может значительно снизить количество ошибок. В результате модемы V.32 обеспечивают очень низкий уровень ошибок и надежный высокоскоростной обмен данными между модемами.

В 1989 году МККТТ предпринял попытку преодолеть технологические барьеры и расширить V.32 до скорости 14400 бит/с. Этот стандарт получил название V.32bis, так как был представлен не как новая идея, а как развитие V.32. V.32bis требует лучшей компенсации эха, чем V.32. Он также требует всеобъемлющего улучшения качества приемника. Испытания однако показывают, что передача со скоростью 14400 бит/с по стандартным телефонным линиям возможна лишь при очень высоком качестве модема.

V.32bis был официально принят МККТТ только в 1991 году.

Стандарт V.42

С приближением скорости передачи данных к теоретически предельной для телефонных линий МККТТ и изготовители модемов ищут новые пути повышения производительности и увеличения скорости передачи. Два наиболее важных шага в этом направлении — рекомендации V.42 для коррекции ошибок и рекомендации V.42bis для сжатия данных.

Функции коррекции и сжатия применяются к данным перед их передачей (модуляцией) и снимаются перед декодированием.

Высокоскоростные модемы делают большое количество ошибок не только из-за затухания сигнала, но и из-за использования для передачи данных верхней и нижней частотных границ диапазона (которые обычно зашумлены сильнее). V.42, принятый в 1988 году, обеспечивает коррекцию ошибок с помощью метода автоматического повторения запросов ARQ (Automatic Repeat reQuest).

При использовании ARQ данные в передатчике группируются в блоки и правильность передачи блока определяется путем расчета контрольных сумм с использованием циклических избыточных проверок CRC (Cyclic Redundasy Check). Концепция CRC уже применялась для обеспечения целостности файлов, передаваемых протоколами вроде XMODEM. Основное различие состоит в том, что V.42 обеспечивает коррекцию ошибок для всего информационного обмена, а не только для передаваемых файлов. Так как метод проверки принятых данных и повторной передачи сбойных блоков поддерживается самим модемом, то он незаметен для пользователя и значительно повышает качество обменных процессов.

Основной недостаток V.42, как и любого метода коррекции ошибок — снижение информационной (результирующей) скорости в случае обнаружения многочисленных ошибок из-за повторной передачи блоков. Однако это существенно только при значительном количестве ошибок. Но даже такое изменение результирующей скорости — это очень малая цена за возможность идентификации и коррекции ошибок.

Модемы, оснащенные V.42, поставляются с конца 1988 года (они также поддерживают V.22bis). Сейчас этот стандарт широко используется в модемах, работающих в соответствии с V.32.

Стандарт V.42bis

Принятый после 1989 года стандарт V.42bis обеспечивает первый "официальный" метод для компрессии и декомпрессии данных в модемах (несколько подобных технологий уже запатентовано в настоящее время. Наиболее известным является метод MNP5 фирмы Microcom).

С V.42bis МККТТ принял технологию, подобную той, которая уже применяется в компьютерном мире для сжатия данных. Это один из вариантов алгоритма сжатия LempelZiv, такой же метод сжатия используется в семействе архиваторов .ARC и .ZIP.

Однако вместо сжатия одних только файлов V.42bis выполняет автоматическую, в реальном масштабе времени компрессию и декомпрессию всей информации, передаваемой между модемами. Это может компенсировать потерю времени, возникающую при работе ARQ рекомендации V.42. При использовании V.42bis появляется возможность сжимать многие виды данных до четырех раз. Преимущество уменьшения времени, требуемого для передачи данных, особенно очевидно при работе с удаленным корреспондентом.

Коэффициент сжатия, который может обеспечить V.42bis, зависит от типа передаваемых данных. Алгоритм сжатия использует повторяющиеся участки, которые заменяются более короткими блоками. Это уменьшает количество символов, необходимых для представления данного набора информации. Большое количество повторений обеспечивает более высокую степень сжатия. С другой стороны, случайный набор данных, не содержащий повторяющихся участков, является практически несжимаемым.

Модемы с V.42bis начали появляться летом 1990 года. Сначала этот метод использовался вместе с V.22bis, а позже с V.32. Все больше модемов с V.32bis поддерживают сжатие V.42bis. Надо также отметить, что V.42bis работает только совместно с V.42, поэтому модемы, не соответствующие V.42, сжимать данные этим методом не будут.

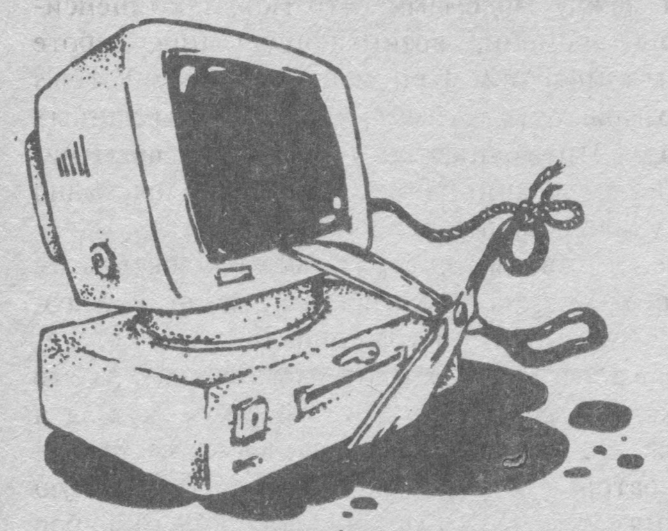
Эпилог

МККТТ продолжает разработку новых стандартов для модемов, каждый раз стимулируя появление новых технологий. Сейчас усилия направлены на стандартизацию скорости 19200 бит/с. Другой стандарт МККТТ, находящийся в развитии, будет обеспечивать единую обменную процедуру, стыкующую между собой различные модемные стандарты. Например, если V.32-модем вызовет V.22bis-модем, новый обменный протокол обеспечит модему V.32 возможность идентификации модемного стандарта корреспондента и возврата в режим V.22bis для совместимой работы. Многие виды модемов уже умеют делать это, но пока нет стандартизованного формата, гарантирующего, что все виды модемов будут делать это одинаково. Новый обменный протокол улучшит совместимость различных типов модемов.

Другим важным стандартом, ожидаемым вскоре от МККТТ, является обеспечение взаимодействия с модемами сотовой сети и модемами для обычных телефонных линий. Интенсивное развитие систем сотовой связи и повышение активности использования портативных компьютеров вне офиса заставит столкнуться с этой проблемой в ближайшем будущем.

И.Гриценюк

По материалам зарубежной печати



Цель данной статьи — описание некоторых приемов защиты программных продуктов с использованием аппаратных особенностей ПЭВМ типа IBM PC.

Железные изюминки защиты

Говоря о защите программной продукции от несанкционированного копирования (НСК), следует отметить все возрастающее значение этой проблемы в связи с развитием и расширением отечественного программного рынка. Разработчики программного обеспечения (ПО) зачастую просто опускают руки перед фатальной неизбежностью пиратского тиражирования своего детища без их на то согласия. Правда, монстры программного рынка, как правило, предпочитают все же не защищать свои программы, так как не несут сколько-нибудь существенных убытков за счет их несанкционированного распространения. Однако для подавляющего большинства фирм и организаций, занимающихся разработкой ПО, объем продаж их (зачастую единственной) разработки является основным средством существования. К тому же, нельзя не учитывать и моральный аспект этой проблемы: нередки случаи изменения названия программы и ее принадлежности фирме-изготовителю. Это приводит к моральным потерям, зачастую не менее тяжелым, чем материальные, так как они бьют по самому больному месту — самолюбию авторов. Поэтому практически всем разработчикам рано или поздно приходится сталкиваться с проблемой защиты ПО от НСК.

При изучении данного вопроса разработчик ПО в первую очередь оказывается перед дилеммой: писать ли защиту самому или же приобрести один из уже существующих пакетов, которых, к слову сказать, сейчас на отечественном рынке программной продукции появилось великое множество. Обычно выбор определяется важностью защищаемой информации, но, пожалуй, решающее значение здесь имеет квалификация программиста. Для профессионала вопрос чаще всего решается в пользу создания собственной защиты, так как большинство ныне существующих пакетов защиты от НСК генерируют так называемый пристыковочный защитный механизм, который, чего греха таить, является чистым надувательством, поскольку легко устраняется, а наиболее опытными хакерами снимается за считанные минуты, даже не удостаиваясь изучения (это может делать, например, утилита CATCHER, рекламируемая центром САПР Ассоциации “Наука“). Кроме того, профессионал никогда не отдаст на откуп другому человеку такую интересную задачу, как создание собственной защиты. Для непрофессионала же выбор чаще всего падает на уже существующие пакеты защиты. Правда, нередки случаи привлечения профессиональных программистов со стороны для написания встроенной защиты, но, поскольку это может быть сопряжено с утечкой информации (от которой как раз и стремится застраховаться пользователь), то такой путь пока не типичен для нашего рынка. Мы, конечно же, сторонники создания собственной встроенной защиты, иначе не взялись бы за написание этой статьи.

Прежде чем перейти к основной теме, позвольте сказать пару слов об общих принципах построения защиты от НСК.

Не будем докучать читателю длинными рассуждениями и скажем только, что на данный момент существуют два типа защиты от НСК: “по дискете“ и “по машине“ (пользуясь жаргоном программистов, занимающихся вопросами защиты ПО). Защитные механизмы первого типа осуществляют проверку характеристик и информации, находящейся на ключевой дискете, которая является, как правило, и носителем защищаемого ПО. Защитные механизмы второго типа осуществляют проверку характеристик компьютера на предмет совпадения их с некоторыми заранее определенными характеристиками разрешенного для работы

с данным программным продуктом компьютера. Для создания более эффективной встроенной защиты наиболее приемлем второй тип защитных механизмов, так как проверка характеристик компьютера достаточно быстра и поэтому может часто повторяться в теле программы, практически не снижая ее быстродействия. Если же ваша программа не работает в реальном режиме времени и вы считаете возможным многократное обращение в программе к такому медленнодействующему устройству, как гибкие диски, то первый тип защитных механизмов также может быть использован для построения встроенной защиты от НСК. Нас же в дальнейшем будут интересовать лишь защитные механизмы второго типа, причем в основном в преломлении к такой сложной, и, на наш взгляд, наиболее интеллектуальной отрасли программирования, как защита программных продуктов от исследования.

Наиболее типичные характеристики компьютера, к которым чаще всего осуществляется “привязка“ защищаемой программы, можно классифицировать следующими тремя группами:

1) временные характеристики:
- быстродействие различных компонент компьютера: процессор, память, контроллеры и т.д.;
- скорость вращения двигателей дисководов;
- время реакции на внешние воздействия;

2) тип микропроцессора (МП) и конфигурация машины:
- тип МП и разрядность шины данных;
- наличие и тип контроллеров внешних устройств (в том числе сопроцессора);

3) характерные особенности компьютера:
- контрольная сумма BIOS;
- содержимое CMOS-памяти;
- длина конвейера шины данных;
- аномальные явления при программировании МП.

Прежде чем описывать приведенные характеристики, хотелось бы отметить, что разделение их на указанные три группы очень условно и чисто субъективно и было произведено нами скорее по схожести способов использования их внутри одной группы для написания защиты, нежели для выделения общности их свойств. Хотя очевидно, что характеристики первой группы удовлетворяют обоим условиям, чего не скажешь о характеристиках последней.

Временные характеристики компьютера, объединенные нами в первую группу, как видно, связаны с быстродействием компонент машины и в наибольшей степени отражают ее индивидуальные особенности. Поэтому их, пожалуй, можно наиболее эффективно применять для “привязки“ ПО к компьютеру. Правда, существует некоторое неудобство при использовании этих свойств в защитных механизмах, которое чаще всего и отпугивает программистов от этой идеи. Имеется в виду нестабильность электронных элементов схемы компьютера и, как следствие, нестабильность показателей быстродействия его компонент, что неизбежно приводит к усложнению защитного механизма,

```
0000              code    segment para public
                          assume cs:code,ds:code

0000              sample1 proc

0000  FA                 cli
0001  BA 0043            mov dx,43h        ; включить 0-й канал
0004  B0 34              mov al,34h        ; таймера
0006  EE                 out dx,al
0007  B2 40              mov dl,40h        ; задать начальное
0009  8A C6              mov al,dh         ; значение счетчика
000B  EE                 out dx,al
000C  EE                 out dx,al

                 ;=========================================
000D  33 F6              xor si,si             ;
000F  8B FE              mov di,si             ; Здесь может быть
0011  B9 8000            mov cx,8000h          ; расположен любой
0014  FC                 cld                   ; ваш текст
0015  F3> A5             rep movsw             ;
                 ;=========================================

0017  BA 0043             mov dx,43h       ; зафиксировать текущее
001A  B0 04               mov al,4         ; значение счетчика
001C  EE                  out dx,al
001D  B2 40               mov dl,40h     ; считать значение счетчика
001F  EC                  in  al,dx      ; в AX
0020  8A E0               mov ah,al
0022  EC                  in  al,dx
0023  86 E0               xchg ah,al
0025  FB                  sti

            ; здесь должна быть проверка считанного значения

0026  B8 4C00             mov ax,4c00h
0029  CD 21               int 21h

002B                      sample1 endp
002B                      code    ends
                  end     sample1
```

Рис. 1

в функции которого должно входить нормирование замеряемых временных показателей. Тем не менее мы все же кратко опишем некоторые методы их измерения.

Наиболее простой в реализации способ измерения быстродействия компьютера представлен в фрагменте программы на рис. 1, который занимается лишь тем, что измеряет количество отсчетов таймера, совершенных за время выполнения указанного участка программы. Результат заносится в АХ и может далее использоваться по вашему усмотрению. Напомним, что непосредственное сравнение полученного результата с некоторой заранее измеренной константой может привести, как уже говорилось выше, к ошибке из-за нестабильности задающего генератора ПЭВМ.

Как вы уже, наверное, догадались, измерения быстродействия как такового в приведенном примере не производится: замеряется лишь время выполнения некоторого заранее выбранного участка программы. На разных компьютерах это время будет скорее всего различным, что и позволяет нам надеяться на работоспособность защищаемого ПО только на определенной ПЭВМ.

Для измерения временных характеристик памяти мы рекомендуем вам также использовать пример, приведенный на рис. 1. Здесь в теле замеряемого участка программы, как вы наверное догадались, осуществля-

ется простая перезапись содержимого области памяти (своего рода регенерация памяти). Как правило, быстродействие одних и тех же участков памяти на разных ПЭВМ различно, что и может быть использовано вашей программой для "привязки" к характерным особенностям компьютера.

Скорость вращения дисководов также может быть определена описанным выше способом (например, измерением по таймеру времени выполнения операции чтения/записи определенного сектора или дорожки, как это делает COPYLOCK), и поэтому мы не будем дольше задерживаться на этом вопросе.

Под временем реакции на внешние воздействия мы понимаем время, требуемое компьютеру на отклик на какую-либо команду внешнего устройства, или же наоборот, время отклика внешнего устройства на команду компьютера. Примером могут служить, например, время установки бита готовности некоторых контроллеров; время, проходящее между выдачей команды в контроллер НГМД и входом в аппаратное прерывание 0eh (и аналогично для других аппаратных прерываний), и др. Измерение времени реакции можно производить способом, описанным выше (по отсчетам таймера), но можно для этой цели использовать и свой программный счетчик, как это делает, например, BIOS при дисковых операциях.

С точки зрения "привязки" защищаемой программы к характеристикам компьютера информация о типе и конфигурации ПЭВМ не является уникальной (особенно если речь идет о компьютерах одной партии), однако может использоваться как дополнительный элемент защитного механизма. Кроме того, знание типа и конфигурации компьютера может быть чрезвычайно полезным, если вы пишете защитный механизм, ориентирующийся на все типы машин.

Информацию о конфигурации можно получить через прерывание BIOS 11h (или, что практически то же самое, по адресу 40:0010 оперативной памяти BIOS), а также из CMOS-памяти, содержащей более исчерпывающую информацию о машине. Гораздо сложнее получить программным путем информацию о типе МП; об этом мы скажем несколько слов при описании последней группы характеристик компьютера.

Характеристики третьей группы, очевидно, не являются (или, точнее, не всегда являются) индивидуальными особенностями некоторого конкретного компьютера. Так, например, общеизвестно, что в пределах одной партии компьютеров контрольная сумма BIOS, как правило, одинакова (если, конечно, речь не идет о компьютерах так называемой "белой" сборки). Энергонезависимая память CMOS, содержащая часы реального времени и информацию о конфигурации машины, также зачастую одинакова на компьютерах одной партии (за исключением байтов, содержащих счетчики часов), разумеется, если они имеют одинаковую конфигурацию. К сожалению, формат CMOS-памяти различен на разных типах машин, что делает "привязку" к ней не универсальной. Если же вы рассчитываете на определенный тип компьютера, то привязка к CMOS вполне возможна, с тем условием, что конфигурация машины не будет меняться в процессе эксплуатации программы.

Напомним, что операции чтения/записи байтов CMOS осуществляются следующим образом:

1) чтение

```
        . . .
        mov al,номер байта CMOS
```

```
        out 70h,al
        jmp $+2
        in al,71h
        . . .
```

2) запись

```
        . . .
        mov al,номер байта CMOS
        out 70h,al
        jmp $+2
        mov al,новое значение
        out 71h,al
        . . .
```

Для “привязки“ к CMOS чаще всего используется следующий алгоритм. Программа-инсталлятор, которая осуществляет установку вашего программного продукта, например, с дискеты на жесткий диск, “накладывает“ по какому-либо закону на определенные участки вашей программы некоторую область CMOS. При запуске установленная программа сама осуществляет обратную операцию, дешифрируя указанные участки своего кода. “Наложение“ можно осуществлять любой приемлемой для этой цели арифметической и/или логической операцией (чаще всего используется “исключающее ИЛИ“), и удобнее всего использовать область CMOS, защищенную контрольной суммой (байты 10h—20h). Но можно использовать и свою собственную информацию, записанную в резервные байты CMOS, если, конечно, формат CMOS позволяет это сделать. Так, например, широко распространенные сейчас компьютеры фирмы WIPRO Information Technology допускают достаточно вольное обращение с CMOS и позволяют осуществлять запись даже в байты, защищенные контрольной суммой (естественно, с последующей корректировкой контрольной суммы в байтах 2eh, 2fh). А такие компьютеры, как SystemPro фирмы Compaq, практически не оставляют лазеек для программиста, но зато обладают (как и некоторые другие компьютеры на базе МП 80386 и выше) таким интересным свойством, как пользовательский пароль, который может быть записан в определенную область CMOS (байты 38h—3fh) и защищен от чтения и записи.

С точки зрения защиты ПО от исследования отладочными средствами очень интересно, на наш взгляд, использование конвейера шины данных МП. На рис. 2 изображен фрагмент текста защитного механизма, генерируемого общеизвестным пакетом защиты COPYLOCK. (Мы надеемся, что не навлечем на себя гнев цивилизованных пользователей отечественного программного рынка, афишируя здесь некоторые тонкости этой программы: ведь она уже достаточно устарела да и давно вскрыта даже самыми ленивыми хакерами. Кстати, настойчиво рекомендуем начинающим специалистам в области защиты программных средств ознакомиться с работой ее защитного механизма, полный листинг которого вы можете найти в электронном журнале “НСК“, № 1´92.) Текст примера специально дан со значением смещений относительно кодового сегмента, чтобы читатель смог увидеть, что команда REP STOSW в подпрограмме SUBR затирает значением в AX-область памяти, в которой находится и сама подпрограмма. Тем не менее, SUBR нормально отрабатывает и возвращает управление в точку вызова, если только не выполнялась в пошаговом режиме под отладчиком.

Трюк здесь очень прост: так как длина конвейера не может быть менее 4-х байт, то, очевидно, команды, расположенные за REP STOSW, уже находятся в нем еще до выполнения этой строковой команды ассемблера. Это и обеспечивает нормальную работу подпрограммы даже после затирания ее собственного кода. Если же выполнение шло по одному шагу — то есть с входом в подпрограмму обработки трассировочного прерывания после каждой команды — то после вы-

```
        . . . . .
cs:07FF         mov     cx,40Eh
cs:0802         mov     di,08a6h
cs:0805         call    subr

                . . . . .

;=========================================
; Затирание указанного участка памяти.

        subr    proc    near
cs:0D63         pushf
cs:0D64         cld
cs:0D65         mov     ax,ds
cs:0D67         mov     es,ax
cs:0D69         rep     stosw
cs:0D6B         popf
cs:0D6C         retn
        subr    endp
```

Рис. 2

```
0000            code    segment para public
                assume cs:code,ds:code

0000            sample2     proc

0000  0E                    push cs
0001  1F                    pop ds
0002  C6 06 0008r 00        mov byte ptr m+1,0   ; изменение
                                         ; смещения в команде JMP
0007  EB 06             m:  jmp short norm_ex
0009  BA 001Br              mov dx,offset trace  ; выполнение с
                                                 ; трассировкой
000C  EB 04 90              jmp exit
000F                    norm_ex:                ; выполнение без
                                                ; трассировки
000F  BA 0026r              mov dx,offset norm
0012                    exit:
0012  B4 09                 mov ah,9
0014  CD 21                 int 21h
0016  B8 4C00               mov ax,4c00h
0019  CD 21                 int 21h

001B  54 72 61 63 69 6E 67+   trace   db `Tracing!`,0ah,0dh,`$`
      21 0A 0D 24
0026  4E 6F 72 6D 61 6C 20+   norm    db `Normal exit.`,0ah,0dh,`$`
      65 78 69 74 2E 0A 0D+
      24

0035               sample2  endp
0035               code     ends
                            end     sample2
```

Рис. 3

полнения команды REP STOSW нарушение исполняемого кода приведет к непредсказуемому результату.

Следующий пример (рис. 3) демонстрирует, на наш взгляд, более изящное использование конвейера. Он позволяет определить, идет ли выполнение программы с трассировкой или нет, и осуществить ветвление (команда JMP с меткой CMD) в зависимости от принятого решения. В примере ветвление используется лишь для индикации принятого программой решения путем выдачи на экран соответствующего сообщения, но вы, естественно, можете использовать его по своему усмотрению для каких-либо более серьезных действий против исследования.

На рис. 4 также изображен пример использования конвейера шины данных, но в несколько иной интерпретации и более завуалированном виде. По существу, он является вариацией на ту же самую тему и лишь демонстрирует разнообразие способов использования конвейера при написании защитных механизмов.

Наиболее интересным нам кажется рассмотрение аномальных явлений, с которыми приходится сталкиваться при программировании МП Intel 80x86. Информацию о них программист чаще всего может получить лишь экспериментальным путем, так как хорошее описание МП этого типа в нашей стране все еще большая редкость, да и там не всегда есть нужные сведения.



Аномалиями программирования называют такие особенности работы компьютера, которые являются исключением из общих правил его функционирования и, как правило, не описаны в популярной литературе. Обычно их очень немного. Исключение здесь составляют уже упомянутые машины фирмы Compaq, которые имеют длинный их перечень, связанный в основном с работой в защищенном режиме и хорошо описанный в технической документации.

Одна из интересных аномалий МП 80x86, о которой уже упоминалось в печати (статья "Особенности работы МП 8086/8088 в пошаговом режиме", Журнал д-ра Добба, №2, 1991), — потеря одного трассировочного прерывания после команд, связанных с пересылкой сегментных регистров типа MOV SEG.REG,R/M и

```
0000             code     segment para public
                          assume cs:code,ds:code

0000             sample3  proc

0000  0E                  push cs
0001  0E                  push cs
0002  07                  pop es
0003  1F                  pop ds
0004  BF 000Cr            mov di,offset m
0007  F9                  stc
0008  FC                  cld
0009  B0 88               mov al,88h          ; заменить код операции
000B  AA                  stosb               ; следующей команды на
                                              ; mov byte ptr m,al
000C  3A 06 000Cr  m:     cmp al,byte ptr m
0010  73 06                 jnc norm_ex
0012  BA 0024r              mov dx,offset trace   ; выполнение с
                                              ; трассировкой, если
0015  EB 04 90              jmp exit          ; бит CF остался
                                              ; установленным
0018                      norm_ex:
0018  BA 002Fr              mov dx,offset norm
001B                      exit:
001B  B4 09                 mov ah,9
001D  CD 21                 int 21h
001F  B8 4C00               mov ax,4c00h
0022  CD 21                 int 21h

0024  54 72 61 63 69 6E 67+    trace    db 'Tracing!',0ah,0dh,'$'
      21 0A 0D 24
002F  4E 6F 72 6D 61 6C 20+    norm     db 'Normal exit.',0ah,0dh,'$'
      65 78 69 74 2E 0A 0D+
      24

003E               sample3  endp
003E               code     ends
                            end     sample3
```

Рис. 4

```
0000             code      segment para public
                           assume cs:code,ds:code

0000             sample4   proc

0000  0E                   push cs
0001  0E                   push cs
0002  1F                   pop ds
0003  17                   pop ss
0004  9C                   pushf
0005  58                   pop ax
0006  F6 C4 01             test ah,1    ; бит TF установлен ?
0009  74 06                jz norm_ex   ; нет -- уйти на norm_ex
000B  BA 001Dr             mov dx,offset trace  ; да -- выполнение
                                                ; с трассировкой
000E  EB 04 90             jmp exit
001                    norm_ex:
0011  BA 0028r             mov dx,offset norm
0014                   exit:
0014  B4 09                mov ah,9
0016  CD 21                int 21h
0018  B8 4C00              mov ax,4c00h
001B  CD 21                int 21h

001D  54 72 61 63 69 6E 67+    trace    db 'Tracing!',0ah,0dh,'$'
      21 0A 0D 24
0028  4E 6F 72 6D 61 6C 20+    norm     db 'Normal exit.',0ah,0dh,'$'
      65 78 69 74 2E 0A 0D+
      24

0037             sample4   endp
0037             code      ends
                           end     sample4
```

Рис. 5

POP SEG.REG. К сожалению, в упомянутой статье эти особенности описаны неполно. Во-первых, для МП 8086/8088 существует еще один тип команд, заставляющих пропускать трассировочное прерывание после их выполнения: MOV R/M,SEG.REG. Во-вторых, для МП выше 8086/8088 также идет потеря трассировки при сегментных пересылках, но только с участием стекового сегментного регистра.

Упомянутое свойство применяется с успехом для определения работы программы под трассировщиком.

Известно, что практически все отладчики при обработке трассировочного прерывания анализируют текущую команду на PUSHF (код 9ch) и сбрасывают при этом Т-бит для предотвращения анализа или запрета трассировки. Поэтому при работе программы под отладчиком последовательностью команд PUSHF, POP AX вы никогда не получите установленный Т-бит в регистре AX. Для "обмана" отладчика перед командой PUSHF можно выполнять одну из указанных выше команд пересылки сегментного регистра (для универсальности лучше стекового). На рис. 5 представлен текст короткой программы, демонстрирующей описанный прием. Команда POP SS заставляет отладчик пропустить следующую за ней команду PUSHF из-за потери трассировочного прерывания и тем самым позволяет программе принять решение о работе под отладчиком (или под любой другой исследующей программой, осуществляющей трассировку).

Образцом знания особенностей работы МП и наиболее лаконичным вариантом распознавания его типа мы считаем подпрограмму, текст которой приведен в статье "Intel insight on specific instructions" (Personal Computer World, April, 1990). Мы не будем приводить здесь текст этой подпрограммы, а лишь кратко упомянем две особенности работы различных типов МП, описанных в ней.

Для определения типа МП выше 8086/8088 в подпрограмме используется то, что МП начиная с 80186 маскируют в счетчиках любых линейных и циклических сдвигов все биты, кроме младших пяти, ограничивая тем самым величину сдвигов 31 битом. Для МП начиная с 80286 характерна следующая особенность: команда PUSH SP заносит в стек значение SP до его декремента в процессе выполнения команды, в отличие от более низших типов МП.

Для пытливого ума эти особенности также могут сослужить хорошую службу при написании защитных механизмов, а именно на таких программистов ориентирована эта статья.

А.Долгин,
М.Потанин



Выбор системы защиты от НСК для программного обеспечения всегда вызывает затруднения. Как преодолеть субъективный фактор при сравнении двух и более систем защиты от НСК между собой? Как оценить надежность защитных механизмов? Именно этим вопросам посвящена данная статья.

Купить или украсть? Оценка защиты

В настоящее время пакеты программ, ориентированные на защиту исполняемых модулей от НСК (несанкционированное копирование) и НСД (несанкционированный доступ), появляются словно грибы после дождя. Если на начало лета 1991 года в мутных потоках несанкционированных копий мы смогли выловить порядка 20 систем защиты [1], то сегодня это число как минимум утроилось. Их пишут почти все, кто умеет считать контрольную сумму, знает струтуру EXE-файла и имеет много свободного времени. Стоимость систем защиты самая разнообразная — от 100 рублей до 30 тысяч рублей. Качество изготовления и возможности тоже самые различные и с ценой пакета практически никак не связаны. Поэтому неискушенному пользователю порой достаточно сложно сделать правильный выбор на рынке подобных продуктов.

Цель данной статьи — выработать подход к проведению экспертизы систем защиты от НСК.

Первый шаг при проведении исследований был направлен на изучение приемов работы хакеров и просто любителей "раздевать" защищенные программы для собственного удовлетворения. Для начала было проведено тестирование 118 программистов, имеющих разную профессиональную подготовку и представляющих различные учреждения, на предмет их взаимодействия с защищенным пакетом. Ответы респондентов на вопросы теста приведены в таблице ниже.

Что интересно, такой стимул, как самоутверждение, ненамного уступает денежному вознаграждению, особенно в условиях нашего специфического рынка. Это и понятно, ведь один из показателей профессионализма разработчика — наличие у него самостоятельно полученных листингов широко известных механизмов защиты, например, пакета COPYLOCK. Поэтому одной из важных характеристик системы защиты является ее популярность. Заниматься исследованиями малоизвестной системы защиты ради самоутверждения никто не будет.

Результаты тестирования показывают, что 98% программистов не будут пытаться преодолевать защитный механизм, если система защиты исполняемого модуля гарантирует:

- невозможность копирования ключевой дискеты широко известными и имеющимися практически у каждого разработчика средствами копирования, такими как комплекс COPY II—PC, COPYWRITE, UNLOCK;
- невозможность исследования программного продукта с помощью широко известных отладчиков, дизассемблеров,

а также, если данная система защиты недостаточно известна.

Ответы на вопросы теста (с учетом знаний архитектуры ПЭВМ, языков программирования, стимула и возможного вознаграждения)

	1	2
Всего тестировалось (человек)	98	20
Вскрывать защиту не стану	58	6
Попытаюсь скопировать защищенную дискету стандартными средствами	40	14
1) из любопытства	40	14
2) из самоутверждения	21	3
3) за вознаграждение	40	13
4) за солидное вознаграждение	40	14
Попытаюсь посмотреть, как она устроена, с помощью стандартных средств (отладчик, дизассемблер)	7	13
1) из любопытства	0	4
2) из самоутверждения	0	5
3) за вознаграждение	3	9
4) за солидное вознаграждение	7	11
Попытаюсь подумать и применить что-нибудь свое, даже если это свое придется разработать	2	5
1) из любопытства	0	1
2) из самоутверждения	0	3
3) за вознаграждение	0	3
4) за солидное вознаграждение	2	5
Применю уже готовый собственный специальный инструментарий	0	2
1) из любопытства	0	1
2) из самоутверждения	0	2
3) за вознаграждение	0	1
4) за солидное вознаграждение	0	2

1. Знание машиннонезависимых языков программирования и умение работать со стандартными утилитами.
2. Хорошее знание ассемблера и архитектуры ПЭВМ.

Таким образом, можно иметь практически полную гарантию надежности системы защиты, если:

- система защиты удовлетворяет названным трем основным требованиям;
- в организации, куда поставляется защищенный продукт, нет талантливых программистов с чувством не полностью реализованных способностей;
- отсутствует широкая известность применяемой системы защиты;
- защищаемый пакет продается дешевле, чем оплачивается работа профессионального хакера.

Как видно из таблицы (последняя часть), только те из программистов попытаются снять защиту с защищенного пакета (удовлетворяющего трем основным требованиям к защитным механизмам), которые имеют собственный оригинальный инструментарий. Таких людей (фирм) на сегодня не так уж и много, но с каждым днем их число увеличивается.

Примерами специальных оригинальных разработок, используемых для "вскрытия" защит, являются:

1. *Эмулятор среды* (подробное описание принципов работы в [1], а также в рекламных материалах Центра "Совмаркет").
2. *Симулятор микропроцессора 8088* (краткое описание принципов работы в рекламных материалах фирмы ЭЛИАС).
3. *Специальный отладчик для работы в защищенном режиме на 386 процессоре* (разработчик просил сохранить анонимность).

На момент публикации статьи названные продукты уже поступили или в ближайшее время поступят в продажу и, вероятно, получат широкую известность, а значит, сделают бесполезными все имеющиеся на сегодня системы защиты. В этом случае основным критерием для оценки надежности системы защиты станет объем защитного механизма. Например, при работе с *Эмулятором среды*, названным выше, разобраться с 5 Кбайт самогенерирующихся команд можно за одну неделю. По субъективным оценкам, объем защитного механизма не менее 30 Кбайт, даже при наличии специального инструментария у хакера, значительно усложняет жизнь исследователя.

До 50% откликов на книгу [1] имеют отношение к экспертизе систем защиты. А чтобы качественно разобраться с каждым защитным механизмом, сравнить их, необходимо время (и много времени). При этом эксперт обязательно вносит в исследовательский процесс свой субъективный фактор, который естественным образом отражается на итоговом заключении.

Это говорит о том, что на сегодня организация квалифицированной объективной экспертизы достаточно актуальна. Учитывая массовость систем защиты от НСК, назрела необходимость стандартизировать проведение экспертиз, то есть создать своего рода АРМ эксперта систем защиты от НСК. Основные функции оператора АРМа заключаются в проведении ряда экспериментов и документальном подтверждении, что исследуемая система защиты не позволяет выполнять над защищенным продуктом определенные действия, в частности:

- копировать ключевые дискеты имеющимися на рынке средствами копирования;
- получать адекватный листинг защитного механизма существующими дизассемблерами;
- исследовать работу защитного механизма существующими отладчиками;
- исследовать работу защитного механизма с помощью специальных программ типа эмуляторов.

Кроме того, в итоговом экспертном заключении должны быть указаны такие характеристики системы защиты, как:

- степень популярности системы защиты (тираж);

- дата изготовления данной версии;
- объем защитного механизма.

Ценность АРМа эксперта систем защиты, как и весомость итогового заключения по конкретной системе защиты, во многом зависит от числа пакетов для копирования, дизассемблирования, отладки, специсследований, включенных в АРМ. Этот критерий позволяет более-менее объективно оценить качество самой экспертизы, что немаловажно. При этом необходимо учитывать и временной фактор, например, по системам защиты экспертное заключение имеет ценность не более 6 месяцев со дня исследования. За этот промежуток времени, как правило, на рынке появляется от 2 до 5 специальных пакетов.

Кроме названных основных характеристик систем защиты, каждая система может выполнять ряд сервисных функций, делающих ее более универсальной и удобной в работе, но не повышающих надежность защиты.

К таким функциям можно отнести следующие возможности пакетов:
- инсталляция с нескольких дискет;
- реинсталляция;
- задание количества инсталляций;
- инсталляция нескольких программ с одной дискеты;
- возможность инсталляции файлов, размер которых больше емкости дискеты;
- привязка при инсталляции с дискеты к аппаратным особенностям ПЭВМ;
- работа без ключевой дискеты;
- индивидуальность защиты;
- установка числа запусков для защищенных программ;
- установка предельной даты;
- использование пароля доступа к программам;
- обнаружение вируса в защищенной программе;
- идентификация и лечение от вирусов;
- визуализация авторских прав;
- допустимый формат дискет;
- защита данных и графических шрифтов;
- возможность привязки к одной ключевой дискете нескольких программ;
- ведение базы данных по созданным ключевым дискетам, защищенным программам и покупателям;
- цена за количество изготавливаемых дистрибутивов.

Таким образом, полное представление об исследуемом пакете защиты можно получить, исследовав все основные и сервисные характеристики пакета. Естественно, что при примерном равенстве в обеспечении надежности товарную ценность пакета определяют его сервисные функции.

В заключение приведем применяемую нами методику расчета надежности системы защиты.

Введем следующие обозначения:

A1 — множество спецсредств, позволяющих автоматически снимать защиту с программ, защищенных с помощью исследуемой системы защиты (например, программы для копирования защищенных дискет, программы типа R_CL (Долгин А.Е. Электронный журнал "НСК", №1. — М.: СОВМАРКЕТ, 1992));

A2 — множество спецсредств, облегчающих пользователю анализ защищенной программы (для исследуемой системы защиты);

r1(t) — вероятность наличия у пользователя средства из множества A1 на момент времени t;

r2(t) — вероятность наличия у пользователя средства из множества A2 на момент времени t;

n1 — вероятность, что пользователь опробует имеющиеся у него средства для автоматического снятия защиты с защищенной программы (согласно таблице, n1=54/118);

n2 — вероятность, что пользователь опробует имеющиеся у него средства для исследования защитного механизма защищенной программы (согласно таблице, n2=20/118);

L — вероятность, что пользователь будет создавать свое средство для автоматического снятия защиты (согласно таблице, L=7/118);

V — объем защитного механизма (в командах);

U — мощность языка программирования (число различных команд языка + специальные приемы программирования);

m1(t) — популярность защищаемой программы (тираж) на момент времени t;

m2(t) — популярность системы защиты (тираж) на момент времени t;

S — стоимость защищаемой программы (в рублях);

Sc — стоимость работы хакера (в рублях);

t — время (число дней).

Тогда вероятность, что защита не будет снята с одной копии защищенной программы ни одним из средств множества A1 на момент времени t, будет определяться по формуле

```
P1(t) = 1-r1(t)+r1(t)*(1-n1)                          /1/
```

С учетом m1(t) копий, соответственно

$$P1(t,m1) = (1-r1(t)+r1(t)*(1-n1))^{m1(t)} \qquad /2/$$

Как было отмечено выше, часто причиной появления спецсредств для снятия защиты является не только цена конкретного защищенного пакета, но и популярность используемой системы защиты. Таким образом, можно констатировать существование зависимости r1(t) от популярности системы защиты и от отношения стоимости работы хакера по снятию защиты к стоимости защищенной программы. Указанная зависимость имеет следующий вид:

$$r1(t) = L^{m2(t)*Sx/S} \qquad /3/$$

Однако защита может быть снята не только с помощью готовых, специально для этого предназначенных пакетов, но и благодаря кропотливому исследованию конкретной защищенной программы.

Вероятность, что хакер не разберется с механизмом защиты, при условии, что он будет разбираться (n2), за t дней, определяется по формуле

```
P2(t) = 1 - exp(-k*t) * n2                            /4/
```

где k — постоянная времени, которая определяется объемом исследуемого защитного механизма, мощностью языка программирования и наличием у хакера средств для исследования пакетов из множества A2, т.е.

```
k = k(U,V,r2)                                         /5/
```

В работе для расчетов использовалась следующая формула определения k, полученная на основании анализа экспериментального материала

```
k(U,V,r2) = -1.8 * V * U /(r2 + 1) + 2.3              /6/
```

В общем виде n2 определяется популярностью пакета и возможным денежным вознаграждением (для расчетов использовались значения n2 из таблицы)

```
n2 = n2(m1,Sx,S)                                      /7/
```

С учетом m1(t) копий P2 приобретает вид

$$P2(t,m1) = (1 - exp(-k*t) * n2(m1,Sx,S))^{m1(t)} \qquad /8/$$

Таким образом, окончательная формула выглядит так

```
P=P1*P2                                               /9/
```

По вышеприведенной методике мы пробовали оценивать надежность широко рекламируемых систем защиты, в частности таких, как SOFTLOCK ("ЭЛКОМ"), SUPER GUARD ("KAMI"), STAS (СП "СЛОТ"), CONVOY ("ЭЛИАС"), DDR ("СОВМАРКЕТ"). Возможно, у нас были не последние версии, но мы для себя сделали следующий вывод: надежность названных систем примерно одинакова и очень сильно зависит от времени.

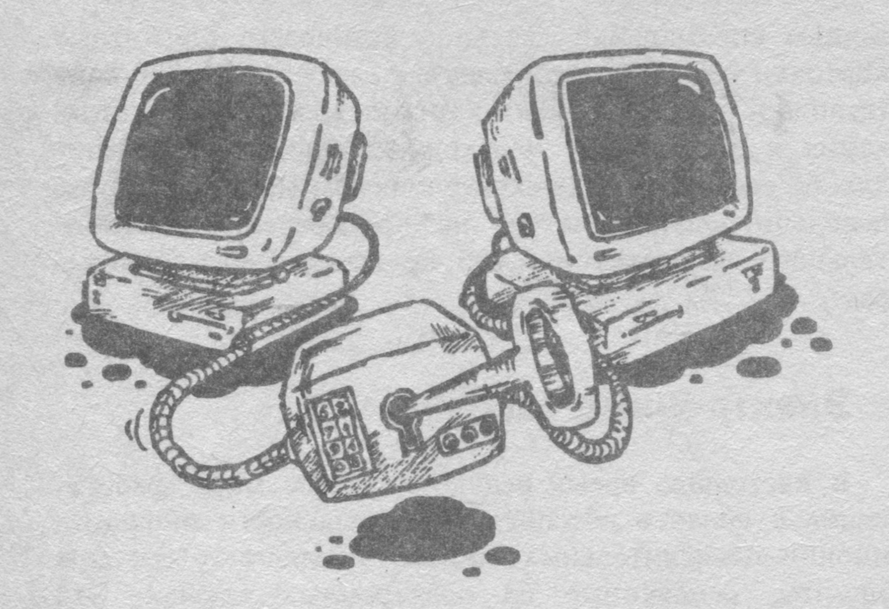
Только достаточно слабое распространение на сегодняшний день таких спецсредств хакера, как программы EXEB, и названных выше эмуляторов не позволяет считать продукты, защищенные указанными системами защиты, открытыми. Для простого пользователя системой защиты все вышесказанное означает, что раз в полгода ему необходимо переходить на следующую версию защитного механизма и к тому же модифицировать собственный пакет, так как старая версия наверняка уже где-то гуляет "раздетая". А что касается выбора системы защиты, то тут подходит следующий алгоритм:

1) ключевая дискета не должна копироваться популярными средствами копирования защищенных дискет, например, COPY II—PC и COPYWRITE;
2) система защиты должна корректно внедрять защитный механизм в исполняемый модуль, то есть защищенная программа должна работать (а это бывает не всегда, так как защищаемая задача может быть оверлейной, писать и читать код и данные из своего тела на диске и иметь другие специфические особенности);
3) из всех систем защиты, удовлетворяющих пунктам 1 и 2, следует выбирать самую дешевую.

С. Расторгуев

Литература:

1. Расторгуев С.П., Дмитриевский Н.Н. Искусство защиты и "раздевания" программ. — М.: СОВМАРКЕТ, 1991.



Оказывается, на ключ можно запереть не только квартиру, сарай или гараж. Электронные ключи позволяют эффективно "запирать" от взломщиков данные и разрабатываемое вами программное обеспечение.

Запирайте программы на ключ

Проблемы защиты информации

По мере все большего распространения персональных компьютеров, а также развития и усложнения средств обработки информации повышается ее уязвимость. Это связано с резким увеличением объемов информации, накапливаемой, хранимой и обрабатываемой с помощью компьютерных систем, сосредоточением в единых базах данных информации различного назначения и различной принадлежности и резким расширением круга пользователей, имеющих непосредственный доступ к ресурсам вычислительной системы. Возрастающий интерес к проблеме защиты информации обусловливается также бурным внедрением вычислительной техники в такие сферы, как биржевое и банковское дело, страхование, медицина и другие.

Основным способом защиты информации является управление доступом к ресурсам вычислительных систем и сетей. Задача сводится к двум частным: защите конфиденциальных данных от несанкционированного доступа и защите программ от незаконного использования (тиражирования и плагиата). Эти задачи решаются комплексом технических, организационных и юридических мер. Технические средства делятся на программные и аппаратные. Защита с использованием чисто программных средств недостаточно надежна, так как всегда можно найти возможность обойти ее. Защита с использованием только аппаратных средств негибка и слишком дорога, поскольку подразумевает применение для решения каждой прикладной задачи специально ориентированного, физически защищенного компьютера. Следовательно, оптимальное решение проблемы должно основываться на комбинации аппаратных и программных средств.

Основные трудности реализации систем защиты состоят в том, что они обязаны удовлетворять двум противоречивым требованиям. С одной стороны, должна обеспечиваться надежная защита информации, находящейся в системе общего доступа. С другой стороны, система защиты не должна создавать неудобств пользователям в процессе их работы.

Для обеспечения системного подхода к защите информации широкое распространение получила стандартизация. В США одним из первых был принят национальный стандарт криптографического закрытия информации — DES, определяющий алгоритм криптографического преобразования информации и правила его использования. Для нашей страны существует аналогичный стандарт — ГОСТ 28147-89.

Специалисты считают криптографическое закрытие информации весьма эффективным как с точки зрения собственно защиты, так и с точки зрения наглядности для пользователей. Криптографическое закрытие является единственным способом надежной защиты информации при ее хранении или передаче по линиям связи.

Стандартный алгоритм шифрования не составляет тайны, ключи же шифрования должны быть доступны только подлинным пользователям. Одна из центральных задач защиты — опознание пользователей; преодоление этого барьера открывает злоумышленнику доступ к информации. Традиционное опознание по паролю недостаточно надежно, — ведь пароль можно подслушать или подсмотреть, перехватить, а то и просто разгадать. В качестве альтернативного унифициро-

3 Зак.№2952

ванного способа используют цифровые коды, генерируемые с помощью некоторой случайной процедуры. Такой код заносится затем на какой-либо носитель информации и считывается в процессе опознания. Преодолеть такую защиту гораздо труднее, чем парольную, в то же время реализовать ее гораздо проще и дешевле, чем системы, использующие индивидуальные характеристики пользователя (отпечатки пальцев, рисунок радужной оболочки глаз, голос, роспись и т.п.).

В настоящее время проблема защиты компьютерных программ от несанкционированного использования становится особенно актуальной. Программное обеспечение (ПО) стало массовым продуктом, начало приносить миллионные прибыли и превратилось в объект большого бизнеса. Наибольший экономический ущерб фирмам — производителям ПО наносят программные пираты — хакеры, которые осуществляют незаконное копирование и реконструирование программ. К сожалению, такая практика получает все большее распространение, особенно в нашей стране, поскольку позволяет извлекать экономическую выгоду от нелегальных продаж, а вероятность применения юридических мер невелика. Серьезные проблемы возникают в связи с повсеместным использованием стандартных накопителей на гибких магнитных дисках. Вследствие применения единых правил записи и считывания, информация, хранимая на гибких дисках, может стать легкой добычей злоумышленников.

По мнению разработчиков ПО, надежная защита должна работать примерно так. Универсальное оборудование программируется уникальным образом. Защищенная программа обращается к нему с некоторыми входными параметрами и получает детерминированные ответы, которые использует в дальнейшей работе. Ответ не должен быть простым — "да" или "нет"; он представляет собой некоторую информационную последовательность — число, причем в программе не должно быть явного сравнения этого числа с эталоном. Чем больший объем информации может вернуть устройство в ответ на обращение, тем лучше. Это могут быть данные или часть исполняемого кода программы; устройство может аппаратно реализовывать какие-либо функции.

Первая мысль по практическому решению этой задачи — разместить соответствующее оборудование внутри компьютера. Однако это потребует вскрытия ПЭВМ и использования одного из разъемов расширения, что не всегда желательно. Более удобным и естественным представляется вариант использования для этих целей внешних устройств, подключаемых к портам ПЭВМ.

В некоторых системах защиты в качестве внешних устройств применяются обычные дискеты, в других — магнитные карточки, аналогичные кредитным. Основной недостаток дискет состоит в том, что записанные на них данные легко скопировать. Иногда для того чтобы сделать дискету уникальной (некопируемой), магнитный слой на ней частично разрушают с помощью специальной лазерной установки. Магнитные карточки имеют небольшую емкость — 256 байт (стандарт для кредитных карточек), а для их считывания к компьютеру необходимо подключить специальное устройство. Подобные способы защиты требуют довольно больших затрат на приобретение дополнительного оборудования, что не всегда себя оправдывает.

Электронные ключи

В настоящее время ведущие фирмы мира, работающие в области обеспечения безопасности информационно-вычислительных систем, предлагают системы защиты, основанные на электронных ключах. Это компактные устройства, которые подключаются к одному из внешних разъемов компьютера, "прозрачные" для периферийных устройств. Наиболее совершенные ключи программируются уникальным образом фирмой-изготовителем или самим пользователем. Электронные ключи зарубежных фирм стоят от 19 до 95 долл. С их помощью можно не только хранить и возвращать информацию по запросу, но и аппаратно реализовывать некоторое ее преобразование. О популярности подобных систем свидетельствует тот факт, что французская фирма MICROPHAR продала с 1983 по 1990 год свыше 500 тысяч электронных ключей, а в 1991 году — около 240 тысяч на сумму 50 млн. франков (около 9,3 млн. долл.). Продукцию MICROPHAR используют более 3200 компаний во Франции и за ее пределами.

1. Электронные ключи MICROPHAR

Ключ собран из электронных компонентов по SMD-технологии (Surface Mounted Device) и залит смолой. Соединение компонентов образует уникальный код, индивидуальный для каждого заказчика (код клиента). Ассемблерная процедура, опрашивающая ключ, различает именно этот код. Если ключ соответствующий, то он возвращает число, относящееся к защищенной программе (код ПО). Ключи с одним и тем же кодом клиента могут возвращать разные числа.

Ключ с памятью подобен электронному ключу, но при этом способен запоминать до 62 байт информации. Тридцать одно 16-разрядное слово может быть адресовано (считано и/или изменено) с помощью специальной процедуры, индивидуальной для каждого клиента.

2. Hardlock (FAST Electronic, Германия, и Glenco Engineering, США)

Hardlock, собранный по SMD-технологии, базируется на специализированной заказной интегральной микросхеме (ASIC), вырабатывающей отклик по спе-

Инструментальный пакет "К л и п к и т" Версия 1.0

Условия поставки

Первая партия пакета в полной комплектации поступит в распространение в конце третьего квартала 1992 г.

Цена пакета - 9000 руб. + 28% НДС.

Распространение пакета осуществляется первоначально по предварительным заказам, а затем в свободной продаже. Оформление предварительного заказа дает право на 20% скидку к цене пакета независимо от количества комплектов в заказе (минимальный объем заказа для организаций - 2 комплекта).

Заказ на приобретение пакета партиями (10 и более штук) дополнительно дает право на следующие скидки к цене пакета:
от 10 до 50 - 10%;
от 50 до 100 - 20%;
от 100 и более - 30%.

По желанию заказчика в комплект поставки может быть включена в соответствии с диллерским договором система Клиппер (Clipper Summer 87 & Clipper Tools One) за отдельную плату в размере 10 000 руб + 28% НДС. При этом указанные выше правила скидок на поставку системы Клиппер не распространяются!

Для оформления предварительного заказа необходимо направить по адресу Корпорации бланк заказа (письмо от организации) с указанием комплектации и количества комплектов, обратного почтового адреса и копии документа о перечислении предоплаты в соответствии с отмеченными выше условиями на первую партию пакета в течение июня-июля 1992 г.

Приобретение партий пакета для последующего распространения осуществляется только на основе заключенного диллерского договора.

С момента поступления предоплаты на счет заказ считается оформленным. Выполнение заказа может быть осуществлено как на территории Корпорации по предварительным приглашениям, высылаемым по адресу заказчика, так и пересылкой пакета почтой.

Для формирования предварительных заказов и более полного информирования потенциальных пользователей о возможностях пакета и технологии разработки программ с его применением в настоящее время подготовлены и распространяются демонстрационный(DEMO) и учебный(TUTOR) комплекты пакета, являющиеся дополнительными составляющими полной комплектации пакета, распространяемыми независимо.

Приобретение этих комплектов позволяет сразу приступить к практическому освоению технологий использования пакета в прикладных разработках и принять обоснованное решение о целесообразности оформления предварительного заказа на полную версию пакета.

Цена демонстрационного комплекта(DEMO) - 300 руб. + 28% НДС.

Цена учебного комплекта(TUTOR) - 600 руб. + 28% НДС.

Приобретение этих комплектов дает право на следующие скидки к цене пакета:
демонстрационный комплект(DEMO) - 10%;
учебный комплект(TUTOR) - 20%.

Зарегистрированные пользователи получают право на приобретение всех последующих версий пакета с 25% скидкой к цене пакета, пользоваться консультациями и методической поддержкой специалистов Корпорации.

В состав полной комплектации пакета входят:

1. Документация (250 стр.) содержащая:
- руководство пользователя;
- справочное руководство;
- учебное пособие по приложениям.

2. Лицензионное соглашение пользователя.

3. Регистрационная карточка пользователя.

4. Четыре дистрибутивные дискеты(типа DS/DD,360K) с архивированной информацией (.LZH), включающие:
Диск #1 - инстал/утилиты;
Диск #2 - библиотеки;
Диск #3 - справочник(NG);
Диск #4 - приложения.

В состав демонстрационного комплекта (Диск DEMO) входят:

1. Документация, которая содержит рекламные проспекты, инструкции, описания примеров, бланки заказа на полный комплект пакета.

2. Одна дистрибутивная дискета (типа DS/DD, 360K) с информацией в архивированном виде (.LZH), содержащей инструкции и программу инсталяции, исполняемые демонстрационные программы (.EXE), исходные тексты программ(.PRG) с комментариями, файлы демонстрационных баз данных и примеров.

В состав учебного комплекта (Диск TUTOR) входят:

1. Документация, которая содержит рекламные проспекты, инструкции, описания примеров, лицензионное соглашение на использование библиотек учебного пакета, бланки заказа на полный комплект пакета.

2. Одна дистрибутивная дискета (типа DS/DD, 360K) с информацией в архивированном виде (.LZH), содержащей инструкции и программу инсталяции, учебную библиотеку пакета с более чем 100 пользовательскими функциями (.LIB), утилиты, исходные тексты демонстрационных программ (.PRG) с комментариями, файлы демонстрационных баз данных и примеров.

ЗАКАЗЫ НАПРАВЛЯТЬ ПО АДРЕСУ: 117607, Москва, Мичуринский пр-т, 31/1, ТМО

БАНКОВСКИЕ РЕКВИЗИТЫ: р/с 006468455 в МАБ "Темпбанк" корр./сч.161104 в РКЦ ГУ ЦБ РФ в г. Москве, МФО 201791

циально разработанному алгоритму, более секретному, чем счетные или переключающие алгоритмы. Hardlock имеет 128 байт памяти: 32 байта доступны из программ для чтения и записи, а 96 байт — только для чтения. В эту часть памяти информация записывается с помощью поставляемого фирмой программатора Crypto-Programmer. Параметры алгоритма соответствуют содержимому памяти и могут быть перепрограммированы. Hardlock выпускается также в виде платы HL-on-Board, вставляемой в слот IBM PC и полностью совместимой с электронным ключом.

3. ACTIVATOR (Software Security, США)

Основой ключа Activator является заказная интегральная микросхема. Кроме этого, Activator/M имеет 128 байт памяти, доступной для чтения и записи из программ. Схема ключа содержит три шестиразрядных счетчика, работающих в режимах прямого и обратного счета. Начальное состояние каждого счетчика программируется фирмой-изготовителем. Каждый счетчик имеет свою схему управления, которая задает направление счета и вырабатывает тактовый сигнал. Сигналы управления появляются при совпадении поступающего на ключ кода со значением, записанным в схеме управления. Это значение программируется фирмой-изготовителем или самим потребителем с помощью специального программатора, поставляемого фирмой Software Security. Появление сигнала на выходе Activator зависит от того, в каком режиме работает ключ. В режиме "И" сигнал переноса должен появиться у всех трех счетчиков, в режиме "ИЛИ" — хотя бы у одного. Режим работы ключа также может программироваться самим пользователем.

4. HASP (ALADDIN Knowledge Systems, Израиль)

Семейство электронных ключей HASP базируется на полузаказных электронных компонентах, что делает копирование схем невозможным. Каждый ключ HASP имеет уникальный внутренний код. Два различных производителя ПО никогда не получат ключи с одинаковым внутренним кодом. Защищенная программа посылает ключу исходный код, а тот, в зависимости от принятого и своего внутреннего кодов, возвращает ответный код, который проверяется программой.

Ключ MemoHASP имеет также 496 байт памяти, доступной для чтения и записи на любых IBM PC или PS/2. Ключ NetHASP разработан специально для работы в сетях. Одного ключа достаточно для активизации защищенной программы с нескольких рабочих станций сети.

5. "Золотой ключик" фирмы АКЛИС

Электронный ключ GoldKey является оригинальной разработкой фирмы AKLIS. GoldKey — это инструмент защиты, который каждый пользователь может применять по своему усмотрению. Фирма AKLIS не скрывает принципы и алгоритмы, положенные в основу своих продуктов, что дает возможность потенциальным клиентам заранее детально ознакомиться с их особенностями и провести экспертные оценки их надежности.

Основная часть схемы ключа представляет собой перепрограммируемое постоянное запоминающее устройство (ППЗУ) емкостью до 256 Кбит. Уникальность ключа, а также функции, которые он реализует, определяются содержимым ППЗУ. В базовом варианте GoldKey комплектуется памятью емкостью 64 Кбит. Применяется энергонезависимая память одного из самых распространенных типов — ППЗУ с ультрафиолетовым стиранием. Могут использоваться другие элементы памяти, выполненные по КМОП-технологии в корпусе DIP28. Микросхема ППЗУ съемная, программируется фирмой-производителем или самим пользователем с помощью большинства выпускаемых программаторов. Производители ППЗУ гарантируют число циклов перепрограммирования не менее 100, время хранения информации — не менее 10 лет. Файлы для "прошивки" ППЗУ создаются с помощью утилит и редактора, поставляемых фирмой. В процессе отладки программы

разработчик имеет возможность менять и перепрограммировать ППЗУ.

Все электронные ключи, рассмотренные выше, реализовывали какой-либо один алгоритм. Некоторые имели дополнительную возможность хранить информацию. Главное отличие GoldKey состоит в том, что в зависимости от каждого конкретного применения это устройство может реализовывать различные функции и алгоритмы. Этим GoldKey напоминает устройство микропрограммного управления. Пользователь получает возможность выбирать функции ключа из некоторого набора, что позволяет ему организовать защиту своей программы необходимым образом.

Вот перечень некоторых функций, которые могут быть реализованы, причем в одном устройстве можно запрограммировать не одну, а несколько функций:
- логические функции;
- сравнение чисел;
- пороговая и мажоритарная функции;
- суммирование;
- счет по произвольному основанию;
- подсчет числа единиц;
- подсчет сигнатур;
- генерация псевдослучайных последовательностей;
- преобразование кодов;
- конвейерное преобразование;

- хранение и передача в ПЭВМ до 4096 байт информации;
- распознавание последовательности символов.

GoldKey — это ряд систем защиты, базирующихся на электронном ключе. Применяются они для защиты данных и программного обеспечения ПЭВМ от несанкционированного доступа и копирования. GoldKey позволяет:
- обеспечить конфиденциальность данных в ПЭВМ;
- не допустить утечки информации по открытым каналам связи;
- защитить программы от несанкционированного использования;
- разграничить доступ к общим ресурсам компьютера.

Защита данных. Первые две задачи решаются с помощью системы GoldKey-Crypto. Защита осуществляется криптографическими методами на основе алгоритма гарантированной стойкости в соответствии с ГОСТ 28147-89. Данные с различными грифами секретности могут храниться и обрабатываться на IBM PC XT/AT/386/486 и совместимых ПЭВМ, работающих под управлением операционных систем MS-DOS или PC-DOS версий 3.0 и выше.

GoldKeyCrypto обеспечивает "прозрачное" использование данных. Это означает, что конфиденциальная информация всегда хранится на магнитных носителях ПЭВМ в зашифрованном виде. После старта системы процесс шифрования/дешифрования производится автоматически, независимо от дальнейших действий пользователя, со скоростью около 25 Кбайт/с для AT286/12 МГц. Состояние (включено/выключено) и настройка (логический диск) системы постоянно индицируются на экране.

Система использует 32-байтные ключи шифрования, которые генерируются случайным образом и заносятся в память GoldKey. Один электронный ключ способен хранить до 21 ключа шифрования. Это позволяет организовать на ПЭВМ иерархическую систему коллективного доступа к общим ресурсам. Электронные ключи простых пользователей содержат один ключ шифрования, а привилегированных — несколько. Администратор, обладающий наивысшим приоритетом, имеет GoldKey с ключами и данными о полномочиях всех пользователей системы.

При передаче секретной информации в сетях и по открытым каналам связи ее получатели должны иметь GoldKey с ключом, на котором она была зашифрована.

Защита программ. Инструментальные средства GoldKeyTools, поддерживающие наиболее популярные языки программирования и СУБД, позволяют разработчикам ПО организовывать разнообразные алгоритмы взаимодействия программ с GoldKey, ограниченные лишь фантазией их авторов. Процедуры обращения программ к GoldKey или вызовы резидентного модуля, выполняющего обращение, необходимо включить в исходные тексты программ перед их компиляцией. Файлы для программирования электронного ключа создаются в соответствии с алгоритмом при помощи редактора GoldKeyEditor. При отсутствии исходных текстов программ специальная утилита GoldKeyInstall позволяет автоматически защитить нужные EXE- и COM-файлы.

Конечный пользователь ПО может делать неограниченное число резервных копий, но работать только с одной, если имеет один ключ.

Надежность защиты во многом определяется самим разработчиком программы и зависит от алгоритма ее взаимодействия с ключом. Защищенные программы не работают с отладчиками, а процедуры обращения к GoldKey защищены от дизассемблирования. Имея образец ключа, взломщик не сможет скопировать его схему в силу особенностей технологии сборки. Возможность считать содержимое ППЗУ не облегчит работы хакеру, так как остальная часть схемы для него недоступна.

AKLIS предлагает ряд совместимых систем защиты, отличающихся уровнем сложности и ценой. Отдельно поставляются дополнительные модули, позволяющие усовершенствовать приобретенную систему защиты.

Е.Федулов
(095)153-32-19, АКЛИС

Использованы материалы:
Хоффман Л.Дж. Современные методы защиты информации. Пер. с англ. — М.: Сов. радио, 1980.
Зарубежная радиоэлектроника, 1989, №12.
BYTE, January, 1991.
Мир ПК, 1991, №4, с. 69-73.
HASP Programmer's Guide.
Рекламные материалы фирмы MICROPHAR.
Рекламные материалы фирмы FAST Electronic.

Словарь терминов Microsoft Windows

Мы продолжаем публикации, посвященные терминологии, используемой в Microsoft Windows. Сегодня мы публикуем словарь, который приводится в книге Д.Рогаткина и А.Федорова "Turbo Pascal в среде Windows".
Предлагаемый вашему вниманию словарь терминов, используемых при работе и программировании в среде Microsoft Windows, является попыткой авторов внести порядок в то многообразие терминов, которыми изобилуют отечественные публикации по данному предмету.
Мы не претендуем на полноту и законченность, а лишь предлагаем свой вариант терминологии Microsoft Windows. По нашему мнению, дискуссия на эту тему должна привести, в конечном итоге, к появлению однозначных русскоязычных эквивалентов английских названий.
Авторы будут благодарны за любые пожелания и критику в адрес приведенных ниже переводов терминов.

A

Accelerator — акселератор, командная клавиша. Комбинация клавиш, приписываемая определенной команде меню. Обычно управляющие клавиши приписываются наиболее часто используемым командам меню.

Active Window — активное окно. Окно, находящееся поверх других окон и получающее все сообщения о вводе с клавиатуры. Активное окно имеет заголовок, выделенный специальным образом.

ANSI Character Set — стандартный набор символов. Кодировка символов, рекомендуемая Американским Национальным Стандартом.

Application — прикладная программа, приложение. Программа, выполняемая в среде Windows.

B

Background — фон. Область, служащая подложкой для отображения различных графических элементов.

Bitmap — графическое (растровое) изображение. Графическое изображение, представленное в виде массива цветовых точек.

Brush — кисть. Ресурс этого типа используется в операциях закраски и заливки областей.

C

Call-Back Function — косвенно-вызываемая функция, функция, вызываемая извне. Функция прикладной программы, вызываемая ядром Windows. Такие функции имеют специальный пролог и эпилог.

Caption Bar — заголовок окна. Специальная область окна, располагаемая в верхней части и содержащая строку текста.

Caret — курсор при редактировании. В отличие от индикатора положения мыши, курсор редактирования используется для индикации положения следующего вводимого символа в строках редактирования.

Character — символ. Один из элементов набора символов (обычно содержащего 256 символов), используемого прикладной программой для отображения информации.

Check Box — кнопка с независимой фиксацией. Обычно такие кнопки организуются в группу, и в группе может быть выбрано сразу несколько кнопок.

Child Window — дочернее окно. Окно, имеющее родительское окно и зависящее от него.

Client — клиент. При организации динамического обмена данными — прикладная программа, запрашивающая данные сервера.

Class — класс. Набор характеристик, задающих общие свойства окон, принадлежащих данному классу.

Class Style — стиль класса. Одно из свойств класса.

Client Area — рабочая область. Область экрана, через которую прикладная программа/приложение производит вывод информации.

Clipboard — область обмена данными. Используется для статического и динамического обмена данными между прикладными программами.

Clipping — отсечение. Механизм, запрещающий отображение графического изображения вне установленных границ.

Color — цвет. Комбинация трех цветовых составляющих, определяющая отображение графического объекта.

Combobox — комбинированный список. Элемент управления, представляющий собой комбинацию списка элементов и строки редактирования.

Control — элемент управления. Интерфейсный элемент, предназначенный для ввода/вывода информации. К таким элементам относятся кнопки различных типов, строки редактирования, списки и т.п.

CUA (Common User Access) — часть спецификации System Application Architecture фирмы IBM, описывающая стандарт пользовательского интерфейса прикладной программы.

Cursor — курсор. Небольшое двухцветное изображение на экране, перемещаемое вслед за мышью или другим указывающим устройством. Используется для индикации точки на экране, на которую будет оказывать действие манипулятор.

D

Device Context — контекст устройства. Структура данных, содержащая физические параметры устройства и используемая для доступа к устройству.

Dialog Box — панель диалога. Специальный тип окна, обычно используемый для отображения и ввода информации.

Discardable Memory — разрушаемая память. Сегмент памяти, который при необходимости может быть удален из памяти ядром Windows.

Dithered Colors — смешанные цвета. Цвета, получаемые в результате растрирования цветов, имеющихся в основном наборе.

Double-Click — двойное нажатие. Нажатие на любую кнопку мыши два раза подряд с малым интервалом.

Dynamic Data Exchange — динамический обмен данными. Организация взаимодействия между прикладными программами, когда одна из них выступает как источник данных (Server), а другая — запрашивает данные (Client). Обмен данными производится через область обмена данными.

Dynamic Link Library — динамически связываемая (компонуемая) библиотека.

Dynamic Linking — динамическое связывание. Установление адресных ссылок на этапе выполнения программы. Механизм динамических вызовов подпрограмм из внешних библиотек.

E

Edit Control — строка редактирования. Строка, предназначенная для ввода/вывода символьной информации с возможностью ее редактирования.

Escapement — наклон шрифта. Угол наклона каждой линии текста, отображаемой данным шрифтом.

F

Far Call — межсегментный вызов. При межсегментном вызове подпрограммы указывается ее полный адрес.

Fixed Memory Segment — фиксированный сегмент. Сегмент памяти, местоположение которого в памяти не может изменяться.

Floating Popup Menu — всплывающее перемещаемое меню. Меню, появляющееся в произвольном месте экрана.

Focus — фокус. Выделение определенного элемента интерфейса, означающее готовность этого элемента к приему информации с клавиатуры.

Font — шрифт. Набор описаний, используемых для визуального отображения символов. Шрифты могут иметь как растровое, так и векторное представление.

G

Global Memory — глобальная память. Область памяти, доступная всем прикладным программам.

Group Box — группа. Ряд интерфейсных элементов, объединенных вместе для удобства работы с ними.

H

Handle — ссылка. Целое число, назначаемое некоторому программному ресурсу, процессу или объекту для его уникальной идентификации при дальнейшем использовании.

Hook — захватчик. Программа, выполняющая перехват сообщений и их предварительную обработку. После такой обработки сообщения передаются адресату обычным образом.

Hungarian Notation — венгерская нотация. Соглашение о наименовании констант, переменных, процедур или функций таким образом, что имя содержит специальный префикс, позволяющий определить тип данных, к которому относится предмет именования.

I

Icon — иконка. Небольшое графическое изображение (пиктограмма) используемое для наглядного определения некоторой функции, состояния и т.п.

Instance — экземпляр. В среде Windows может быть запущено несколько копий прикладной программы. Экземпляр позволяет однозначно охарактеризовать одну из копий прикладной программы.

K

Keystroke — нажатие клавиши. Событие, в результате которого прикладная программа получает сообщение о нажатии клавиши на клавиатуре.

L

List Box — список элементов. Обычно такой список содержит символьные строки или/и графические изображения.

Local Memory — локальная память. Память, принадлежащая конкретной прикладной программе и недоступная другим прикладным программам.

Logical Font — логический шрифт. Шрифт, создаваемый прикладной программой на основе одного из физических шрифтов, известных системе (предопределенных в среде).

M

MDI (Multiple Document Interface) — многодокументный интерфейс. Стандарт на интерфейс, позволяющий окну управлять несколькими дочерними окнами.

Memory Management — управление памятью. Совокупность процедур, обеспечивающих распределение оперативной памяти между задачами и в пределах одной задачи.

Menu — меню. Совокупность элементов, из которых можно делать выбор при помощи устройств ручного ввода.

Message — сообщение. Структура данных, несущая информацию о некотором событии. Содержит в себе код, идентифицирующий событие и другую информацию, однозначно определяющую данное событие.

Message Box — панель сообщений. Частный случай панели диалога, используемой для вывода информации.

Message-Driven Architecture — архитектура, управляемая событиями. Программы, работающие в средах с такой архитектурой, имеют модуль обработки событий и соответственно цикл обработки сообщений.

Message Loop — цикл обработки сообщений. Цикл, выполняющий прием сообщения, трансляцию сообщения и передачу его процедуре обработки сообщений.

Metafile — метафайл. Механизм, представляющий собой средство для запоминания графических изображений без вывода их на конкретное устройство.

Metric — метрика. Количественная информация о размерах отображаемых объектов.

Modal Dialog — модальная панель диалога. Панель диалога, которая, в отличие от немодальной панели, получает все

события, происходящие в системе, даже если она не является адресатом этого события.

Modeless Dialog — немодальная панель диалога.

Moveable Memory Segment — перемещаемый сегмент. В отличие от фиксированного сегмента, местоположение перемещаемого сегмента в памяти может изменяться.

MS-DOS Applications — прикладная программа или приложение, требующее для своей работы только ОС MS-DOS. Может выполняться в среде Windows специальным образом.

Multitasking — многозадачность. Возможность выполнения нескольких задач в параллельном или псевдопараллельном режиме, поддерживаемая операционной средой.

N

Near Call — внутрисегментный вызов. При внутрисегментном вызове подпрограммы указывается ее смещение внутри сегмента.

O

OEM Character Set — набор символов производителя ЭВМ. Обычно содержит ряд символов, уникальных для страны-производителя.

OLE (Object Linking and Embedding) — связывание и включение объектов. Протокол, определяющий взаимоотношение объектов различных прикладных программ при их компоновке в единый объект/документ.

Outline Font — контурный шрифт.

Own Control — элемент управления, определяемый пользователем. Пользователь сам определяет начертание и поведение такого элемента управления.

P

Parent Window — родительское окно. Окно, породившее данное окно и управляющее им.

Pascal calling sequence — соглашение о вызовах языка Pascal. В отличие от соглашения о вызовах языка С, параметры при вызове подпрограммы передаются в стек слева направо, и подпрограмма сама очищает стек по завершении работы.

Pen — карандаш. Ресурс этого типа используется для отображения точек и линий.

Point — точка. Координата экрана в некоторых принятых в системе единицах измерения.

Popup — всплывание. Тип отображения окна, меню или другого элемента интерфейса: появление на экране поверх других элементов.

Program Entry Point — точка входа в программу. Адрес, с которого начинается выполнение программы. (Адрес первой исполняемой инструкции в программе.)

Push Button — кнопка (кнопка без фиксации). Элемент интерфейса, имеющий форму кнопки.

R

Radio Button — кнопка с зависимой фиксацией. Обычно такие кнопки организуются в группу и в группе может быть выбрана только одна кнопка. Английское название выбрано по аналогии с кнопками выбора диапазона волн радиоприемника, когда в каждый текущий момент может быть выбрана только одна волна.

Rectangle — прямоугольник. Прямоугольная область экрана в некоторых принятых в системе единицах измерения.

Region — область. Некоторая область экрана, используемая для вывода информации.

Resource — ресурс. Совокупность данных, включаемая в состав исполняемого модуля прикладной программы. Среда Windows предоставляет функции для доступа к ресурсам.

S

Scroll Bar — полоса прокрутки. Вертикальная или горизонтальная полоса с передвигающимся бегунком посередине и кнопками-стрелками по краям. Предназначена для позиционирования окна просмотра внутри большого текста или для визуального ввода количественной информации (например, интенсивность того или иного цвета).

Server — сервер. При организации динамического обмена данными — прикладная программа, выступающая как источник данных.

Shift State — состояние клавиши переключения регистров (Shift).

Solid Color — цвет основного набора.

Static text — статический текст. Отображаемый объект, не обрабатывающий сообщения.

String — строка. Ресурс, содержащий текстовую информацию, доступ к которому возможен через специальный идентификатор.

System Menu — системное меню. Меню, наличие которого у окна отображается специальной кнопкой в левом верхнем меню окна. Обычно содержит команды для изменения размеров окна, закрытия окна и переключения на другую задачу.

T

Tab Stop — шаг табуляции. Число символов, на которое перемещается курсор при нажатии клавиши TAB.

Template — шаблон. Используется при описании ресурсов типа "панель диалога".

U

User-Defined Button — кнопка, определяемая пользователем. Пользователь задает начертание и поведение кнопки.

V

Virtual Key Code — виртуальный код клавиши. Предписываемые клавишей действия в пределах приложения/прикладной программы.

W

Window — окно. Прямоугольная область экрана, выделяемая прикладной программе для осуществления операций ввода/вывода.

Window Procedure — оконная процедура. Процедура, обслуживающая определенное окно и обрабатывающая сообщения, посылаемые этому окну. Оконная процедура вызывается ядром Windows.

Д.Рогаткин,
А.Федоров

Каталог продуктов фирмы NOVELL

ОПЦИИ СЕТЕВОЙ СВЯЗИ

LAN WorkPlace for DOS v4.0

Обзор

Версия 4.0 LAN WorkPlace for DOS обеспечивает пользователям DOS и Microsoft Windows 3.0 прямой доступ к системам UNIX, ЭВМ VAX фирмы DEC, большим ЭВМ фирмы IBM и другим вычислительным ресурсам, используя набор протоколов TCP/IP, являющийся промышленным стандартом. При создании этого продукта Novell особое внимание обращала на простоту использования, экономное использование памяти, высокую производительность и надежность работы в среде NetWare.

LAN WorkPlace for DOS защищает капитальные вложения заказчиков в оборудование ЛВС, поскольку использует технологию Open Data-Link Interface (ODI). Наряду с существующими адаптерами Ethernet, Token-Ring и Arcnet, ODI поддерживает транспортные протоколы TCP/IP и SPX/IPX. Это позволяет пользователям NetWare взаимодействовать с прикладными программами, выполняющимися на системах UNIX или других системах, поддерживающих TCP/IP, без всякого ущерба для NetWare.

Элементы TCP/IP, имеющиеся в LAN WorkPlace for DOS, также позволяют клиентам NetWare напрямую подсоединяться к серверам NetWare v3.11 в среде, допускающей использование только протоколов TCP/IP.

Характеристики

- Передача файлов. Прикладная программа клиента и сервера FTP (File Transfer Protocol) позволяет пользователям вычислительных систем, поддерживающих FTP, обмениваться файлами. UNIX-подобная командная утилита "RCP" (remote copy — удаленное копирование) включена в качестве альтернативного метода передачи файлов. Также включены утилиты TFTP и TFTPd.
- Эмуляция терминала по протоколу Telnet. Включает эмуляторы терминалов DEC VT100 и VT220 для DOS и Windows. Эти эмуляторы могут одновременно обеспечивать до 10 сеансов с несколькими центральными ЭВМ, поддерживая при этом большинство международных наборов символов клавиатуры и алфавитов в режиме переопределения. Интерфейс TELAPI (Telnet Application Programming Interface) позволяет пользователям применять программы связи и эмуляции терминала других поставщиков для связи по протоколу Telnet с центральными ЭВМ, оснащенными TCP/IP.
- Удаленное выполнение команд. UNIX-подобные команды, включающие "RSH" (remote shell — удаленная оболочка) и "rexec" (remote execute — удаленное выполнение), позволяют пользователю инициировать выполнение программ на удаленных системах, вводя команду DOS, и видеть результаты их выполнения на своем экране.
- Удаленная печать. Команда "RPR" обеспечивает гибкий метод пересылки файлов для печати на центральных ЭВМ UNIX и VMS.
- Обеспечение поддержки прикладных программ NetWare и SPX/IPX через сети TCP/IP посредством транспортировки данных клиента DOS в среде TCP/IP и использования драйвера ODI, который

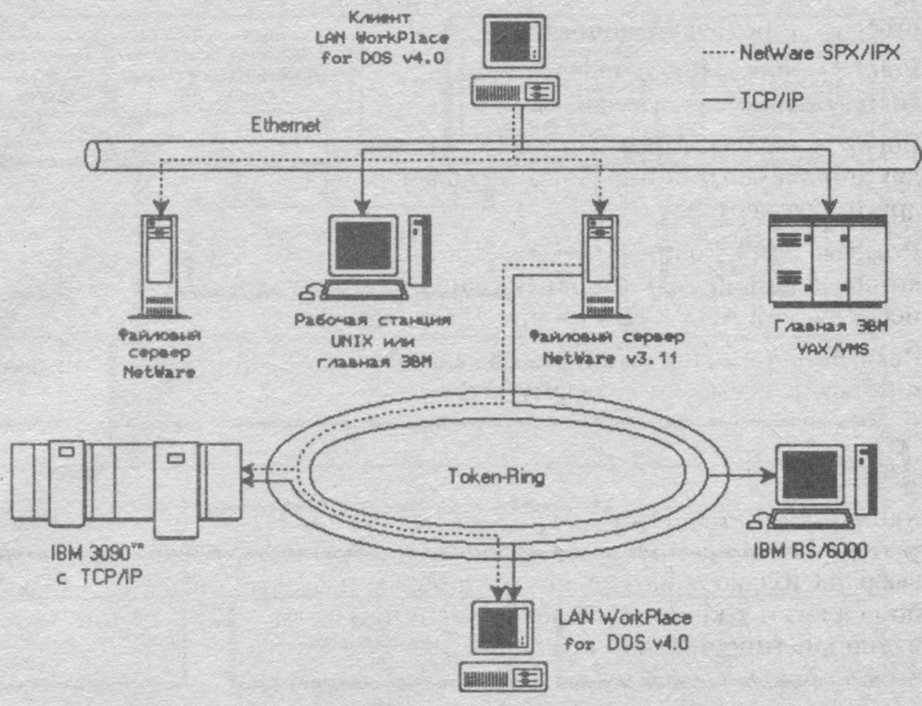


Рис.3.17. LAN WorkPlace for DOS v4.0

Материал подготовлен к публикации совместно СП "ИнтерПроКом" (Тел.: 129-80-09, 128-80-33) и агентством "КомпьютерПресс". Начало в №№3-7´92.

позволяет упаковывать пакеты IPX в пакеты TCP/IP.

Поддерживает передовые сетевые прикладные программы TCP/IP ведущих поставщиков программного обеспечения: DESQview/X фирмы Quarterdeck, X-Vision (X-Server на основе Windows) фирмы VisionWare, SQL Net TCP/IP for MS-DOS фирмы Oracle и сетевую библиотеку SYBASE PC Net-Library for DOS.

- Поддерживает многозадачную среду. Все обычные прикладные программы DOS правильно выполняются под Microsoft Windows или под DESQview/X фирмы Quarterdeck.
- Гибкое использование памяти. Комбинированные сетевые драйверы TCP/IP и ODI занимают менее 45 Кбайт основной памяти, а IPX и оболочка NetWare требуют только 16 Кбайт дополнительной памяти на машинах, имеющих расширенную память. Все сетевые драйверы могут быть динамически загружены или выгружены из памяти без необходимости перезагрузки сервера (см. рис. 3.18).
- RFC-1001/RFC-1002 NetBIOS API (реализация B-node). Этот дополнительный стандартный интерфейс уровня сеансов NetBIOS поддерживает прикладные программы NetBIOS через TCP/IP.

Прикладные программы Windows

LAN WorkPlace for DOS включает три новые прикладные программы TCP/IP, предоставляющие сетевые службы и написанные для Windows 3.0:

- эмуляция терминала. Служба Host Presenter обеспечивает одновременную поддержку до 10 окон сеансов с несколькими центральными ЭВМ, оснащенными TCP/IP, по протоколу Telnet (эмулируя терминалы VT220, VT52 или VT100). Она также поддерживает передачу данных из сеанса Telnet и в него через Windows Clipboard и позволяет просматривать текст, пользуясь зоной "прокрутки" на экране дисплея. Host Presenter также включает хорошо развитый язык написания сценариев, называемый Script Director. Пользуясь этим языком, можно создать более удобный интерфейс для традиционных прикладных программ центральных ЭВМ, тем самым увеличивая и сохраняя их ценность;
- служба Serving FTP выполняется как фоновая задача Windows. Если какой-либо пользователь FTP хочет установить связь с данным персональным компьютером, служба инициирует передачу данных на локальный диск (или том NetWare, к которому этот персональный компьютер имеет доступ) или с него. Запрос на FTP-доступ может осуществляться по паролю, в то время как права доступа к файловой системе NetWare, содержащиеся в бюджете пользователя, ограничивают доступ ко всем дисководам, подключенным к сети NetWare;
- передача файлов. Служба File Express обеспечивает простой графический интерфейс для протокола File Transfer Protocol (FTP), который известен сложным пользовательским интерфейсом. File Express делает копирование файлов и директорий с центральной системы UNIX на том NetWare предельно простым. Еще одним уникальным свойством этой программы является возможность передачи файлов между двумя удаленными ЭВМ, оснащенными службой Serving FTP, без необходимости промежуточного копирования на локальный компьютер или сервер NetWare.

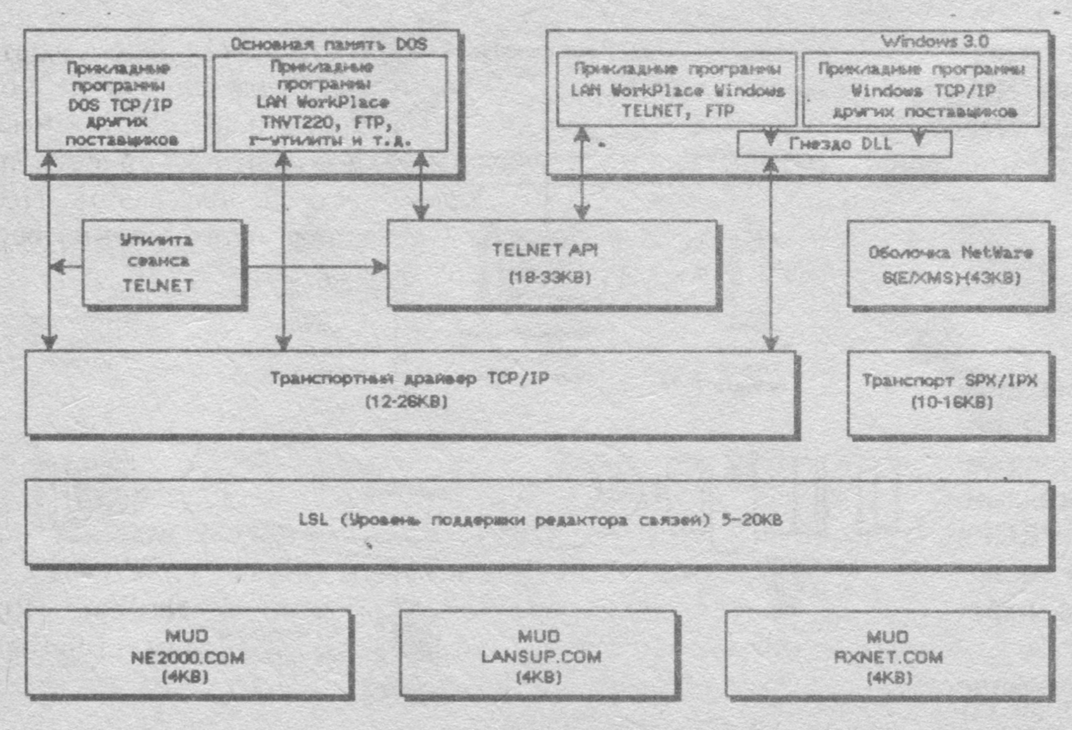


Рис.3.18. Архитектура DOS ODI

Реализация

LAN WorkPlace for DOS v.4.0 использует технологию интерфейса Open Data-Link Interface (ODI) для одновременной поддержки протоколов TCP/IP и NetWare SPX/IPX в сетях Ethernet, Token-Ring, и Arcnet. Узлы сети LAN WorkPlace должны быть подсоединены к тому же участку сети, что и центральная ЭВМ, оснащенная TCP/IP, или находиться на участке сети, соединенном с ЛВС центральной ЭВМ через один или несколько мостов TCP/IP.

Поскольку NetWare v3.11 может функционировать в качестве моста IP, LAN WorkPlace может использоваться вместе с NetWare для 386 процессора для обеспечения связи клиентов сетей Ethernet и Token-Ring с главными системами TCP/IP в участках сети Ethernet (см. рис. 3.16). Мосты IPX в NetWare для 286 процессора не поддерживают маршрутизацию IP и поэтому не должны изолировать узлы LAN WorkPlace от центральной ЭВМ TCP/IP.

Администраторы сети, предполагающие установить LAN WorkPlace for

DOS, должны быть знакомы с основами сетевой организации TCP/IP. Они также должны иметь в виду, что требования прикладных программ, связанных с TCP/IP, могут выходить за рамки возможностей LAN WorkPlace for DOS. Многие поставщики программного обеспечения предлагают прикладные программы, обеспечивающие следующие возможности:
- прямой доступ к файловой системе NFS на центральных ЭВМ, оснащенных TCP/IP, с одновременным доступом к соответствующим томам NetWare;
- участие в распределенных прикладных программах баз данных с доступом к SQL-серверам баз данных, выполняющимся на центральных ЭВМ, оснащенных TCP/IP;
- доступ к приложениям X-Window с использованием персонального компьютера в качестве сервера X-Window;
- эмуляция рабочей станции 3270 по протоколу Telnet.

Необходимое аппаратное обеспечение

LAN WorkPlace for DOS работает на IBM PC XT, AT, PS/2 и совместимых ПЭВМ, а также на ЭВМ с процессором 80386 с минимальным объемом ОЗУ 512 Кбайт. Поскольку Microsoft Windows 3.0 требует для своей работы дополнительной памяти, для использования прикладных программ Windows в рамках LAN WorkPlace for DOS требуется больший объем памяти. Использование памяти детально показано на рис. 3.18.

Также требуется использование адаптера Ethernet, Token-Ring или Arcnet с соответствующим протестированным фирмой Novell драйвером. Поставляется набор протоколов SPX/IPX, а также следующие адаптеры ЛВС и драйверы к ним:

Novell/Anthem	3Com	IBM
NE1000	3C501	Token-Ring
NE2000	3C503	Token-Ring using IBM
NE2	3C505	LAN Support
NE2-32	3C523	
RX-Net		
EXOS 205T		
EXOS 215T		

Необходимое программное обеспечение

LAN WorkPlace for DOS v.4.0 предназначен для работы в операционной системе DOS версий не ниже 3.1 и с Microsoft Windows версий 3.0 и выше.

Спецификации

LAN WorkPlace for DOS поддерживает следующие стандарты протокола Internet Protocol Министерства обороны США:

IP (Internet Protocol)	RFC 791
IP Subnet Extension	RFC 950
IP Broadcast Datagrams	RFC 919
IP Broadcast Datagrams with Subnets	RFC 922
TCP (Transmission Control Protocol)	RFC 793 (0-64 гнезд)
UDP (User Datagram Protocol)	RFC 768 (0-32 гнезд)
IP-IEEE (Internet Protocol on IEEE 802)	RFC 1042
NetBIOS (NetBIOS Service Protocol on TCP/IP)	RFCs 1001, 1002
ARP (Address Resolution Protocol)	RFC 826
RARP (Reverse Address Resolution Protocol)	RFC 903
ICMP (Internet Control Message Protocol)	RFC 792
DNS (Domain Name System) Resolver	RFCs 1034, 1035
TELNET (Telnet Protocol)	RFC 854
FTP (File Transfer Protocol)	RFC 959
TFTP (Trivial File Transfer Protocol)	RFC 783

Информация для заказа

Продукт	Номер изделия
LAN WorkPlace for DOS	
Пакет для одного пользователя	883-001132-001

LAN WorkPlace for OS/2

LAN WorkPlace for OS/2 v2.0 обеспечивает пользователям OS/2 прямой доступ к ресурсам компьютеров Apple Macintosh, систем UNIX, VAX и больших ЭВМ фирмы IBM, используя набор протоколов TCP/IP, являющийся промышленным стандартом. При помощи TCP/IP пользователи могут устанавливать связь своих терминалов с центральными системами, удаленно выполнять программы, передавать и распечатывать файлы на центральных ЭВМ.

LAN WorkPlace for OS/2 позволяет использовать многозадачные возможности OS/2 для одновременной поддержки нескольких сеансов с множеством компьютеров и служб в глобальной вычислительной сети по протоколам TCP/IP. Кроме того, использование NetWare Requester for OS/2 позволяет одновременно осуществлять связь с серверами NetWare.

Характеристики

- Независимость от аппаратных средств и среды передачи предоставляет пользователям широкий выбор топологий (толстый или тонкий кабель Ethernet, витая пара, Token-Ring и т.д.) и сетевых адаптеров.
- Может использоваться совместно с NetWare Requester for OS/2, что позволяет рабочим станциям OS/2 осуществлять доступ одновременно к средам TCP/IP и NetWare.

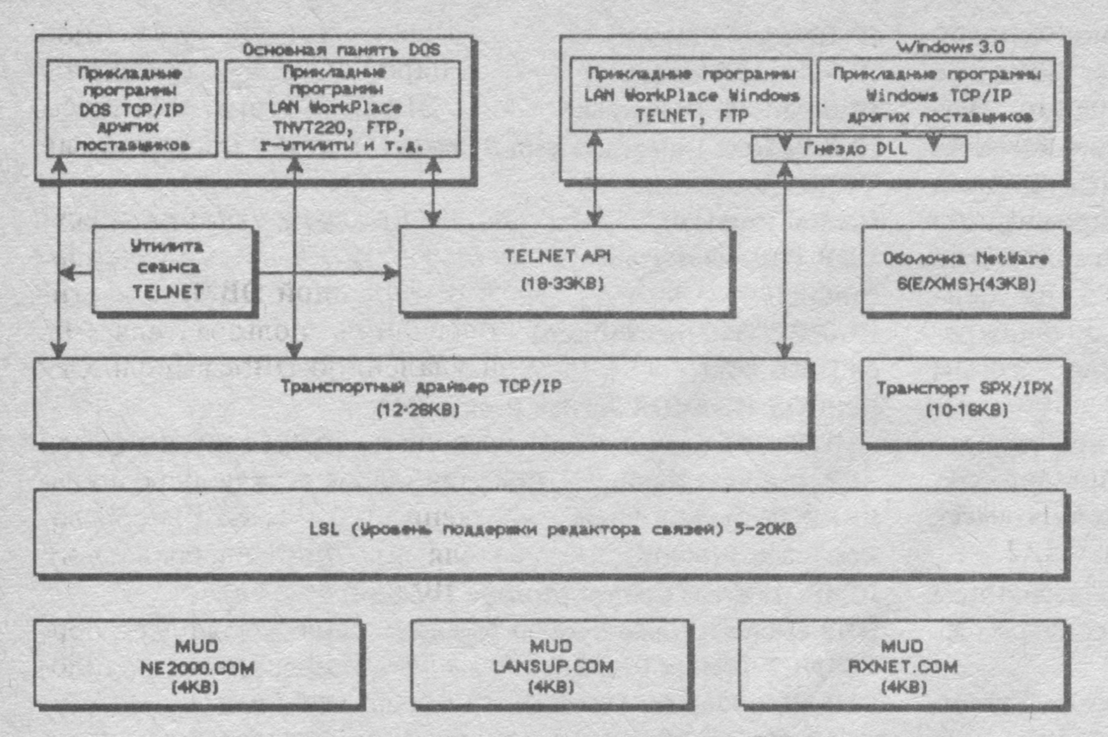


Рис.3.19. Архитектура DOS ODI

- Размещается в ОЗУ, что позволяет использовать данный продукт с любым стандартным сетевым интерфейсом персонального компьютера, поддерживающим спецификацию Network Device Interface Specification (NDIS).
- Включает драйверы NDIS для следующих адаптеров:
  Адаптеры Ethernet EXOS 205T, 205L и 215T
  Адаптеры IBM Token-Ring I и II
  Адаптеры Ethernet 3Com 501 и 503
- Утилиты FTP, FTPD, TFTP и RCP обеспечивают прямой равноправный обмен файлами с любыми системами, поддерживающими TCP/IP и эти прикладные программы.
- Обеспечивает подсоединение терминалов посредством TELNET, DEC VT100 и VT220.
- Транспортные драйверы TCP/IP поддерживают распределенные прикладные программы, написанные для интерфейса взаимосвязи процессов BSD Sockets (такие как X-Window, Oracle и Sybase).
- Поддерживает NetWare для 286 и 386 процессоров, а также серверы SMB и LAN Manager.

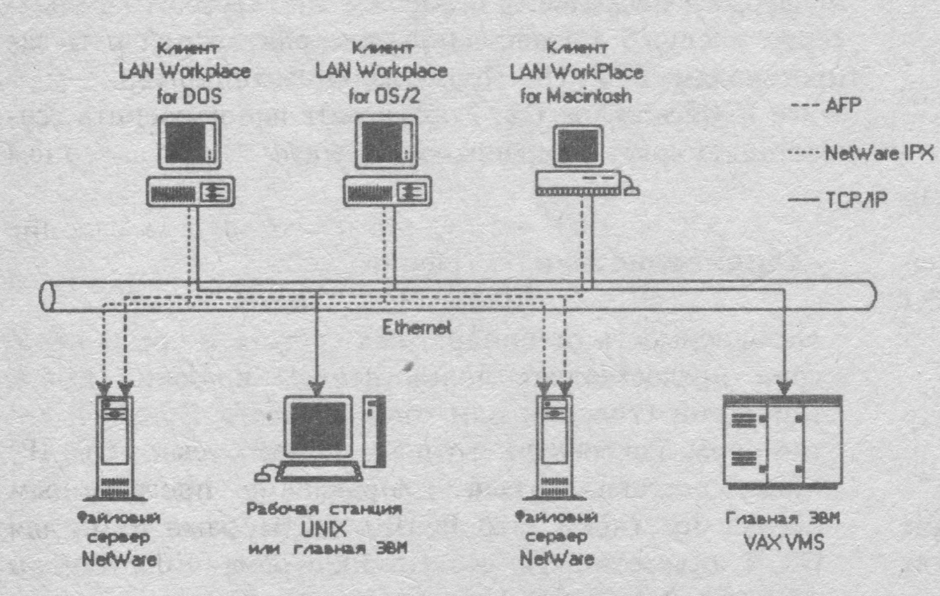


Рис.3.20. LAN WorkPlace for OS/2

- Поддерживает следующие межсетевые протоколы Агентства по перспективным исследованиям (DARPA) Министерства обороны США:
  Internet Protocol (IP)
  Transmission Control Protocol (TCP)
  User Datagram Protocol (UDP)
  Address Resolution Protocol (ARP)
  Reverse Address Resolution Protocol(RARP)
  Internet Control Message Protocol (ICMP)
  Domain Name Service (DNS)
  Telnet Protocol (TELNET)

Реализация

LAN WorkPlace for OS/2 v2.0 предназначен для сетей, в которых необходимо осуществлять доступ с ПЭВМ PC и PS/2, использующий OS/2, к центральным вычислительным ресурсам по протоколам TCP/IP. LAN WorkPlace for OS/2 позволяет клиентам OS/2 NetWare осуществлять доступ к общим ресурсам, используя протоколы TCP/IP. Ресурсы TCP/IP могут включать широкий спектр центральных систем, таких как ЭВМ Sun, IBM или DEC VAX и другие UNIX-системы. Пользователи OS/2 сети NetWare, которым требуется доступ к центральным системам DEC VAX и UNIX, могут использовать LAN WorkPlace for OS/2 для одновременного доступа к этим центральным системам и серверам NetWare.

Перед покупкой LAN WorkPlace for OS/2 менеджер сети должен определить сетевые потребности пользователей OS/2. Данный продукт отвечает следующим требованиям:

- обеспечение машин OS/2 возможностями эмуляции терминала для доступа к центральным системам;
- обеспечение машин OS/2 возможностью доступа клиента к распределенной файловой системе и возможностью обмена файлами между центральными системами и клиентами;
- обеспечение машин OS/2 возможностью одновременного доступа клиента к серверам NetWare и центральным системам TCP/IP;
- обеспечение машин OS/2 возможностью доступа клиента к распределенным прикладным программам, таким как распределенные СУБД, находящиеся на центральных системах VAX и UNIX.

Необходимое аппаратное обеспечение

LAN WorkPlace for OS/2 может выполняться на любой ПЭВМ PC или PS/2 с процессором 80286 или 80386. Для выполнения требуется также сетевой адаптер Ethernet или Token-Ring

с соответствующим драйвером NDIS. В комплект включаются драйверы для адаптеров EXOS серий 205 и 215, адаптеров 3Com 501 и 503, адаптеров IBM Token-Ring I и II. Для использования других адаптеров Ethernet или Token-Ring потребуется соответствующий драйвер, поставляемый производителем адаптера. Отдельный адаптер требуется при использовании продукта NetWare Requester for OS/2.

Необходимое программное обеспечение

LAN WorkPlace for OS/2 требует использования OS/2 версии 1.1 и выше. Может использоваться вместе с продуктом Novell NetWare Requester for OS/2.

Дополнительные элементы

Инструментальный набор Developer's Toolkit, входящий в комплект поставки LAN WorkPlace for OS/2, содержит интерфейс прикладного программирования в виде библиотеки программных гнезд, позволяющий другим разработчикам ПО создавать адаптационные сетевые прикладные программы для среды TCP/IP. Он включает набор обращений, поддерживающих драйверы устройств и позволяющий прикладным программам осуществлять доступ к сети, используя гнездовую модель 4.3 BSD, включающую гнезда потоков и дейтаграмм (массивов данных). Интерфейс этой библиотеки программных гнезд поддерживает Microsoft C.

Спецификации (Утилиты доступа к центральным системам)

Передача файлов

- FTP (File Transfer Protocol) позволяет осуществлять обмен одним или несколькими файлами между персональным компьютером и удаленной ЭВМ. FTP использует пароли для опознания пользователей удаленной ЭВМ и требует, чтобы на удаленной ЭВМ выполнялся программный сервер.
- TFTP (Trivial File Transfer Protocol) позволяет осуществлять обмен одиночными файлами между персональным компьютером и удаленной ЭВМ. Эта утилита не производит опознания пользователя, но также требует, чтобы на удаленной ЭВМ выполнялся программный сервер.
- RCP (Remote Copy) позволяет осуществлять обмен одним или несколькими файлами между персональным компьютером и удаленной ЭВМ. RCP производит опознание пользователя по имени персонального компьютера и удаленной ЭВМ.
- FTPD (File Transfer Protocol daemon) позволяет персональному компьютеру работать в качестве файлового сервера, так что пользователи сети могут осуществлять доступ к файлам этого компьютера, пользуясь утилитой FTP.
- TFTPD (Trivial File Transfer Protocol daemon) позволяет персональному компьютеру работать в качестве файлового сервера, так что пользователи сети могут осуществлять доступ к файлам этого компьютера, пользуясь утилитой TFTP.

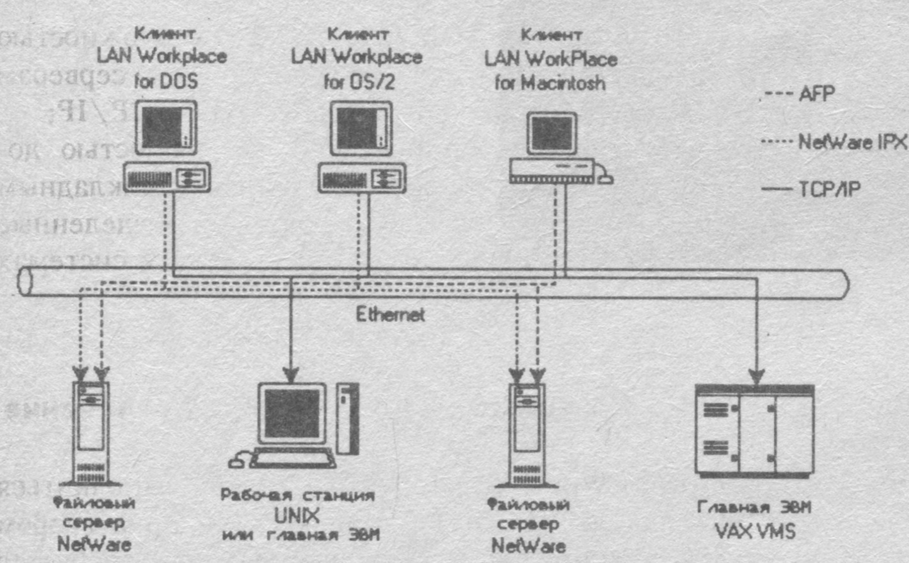


Рис.3.21. LAN WorkPlace for Macintosh

Эмуляция терминала

- TNVT220 позволяет эмулировать на персональном компьютере терминалы DEC VT220, VT100 или VT52.

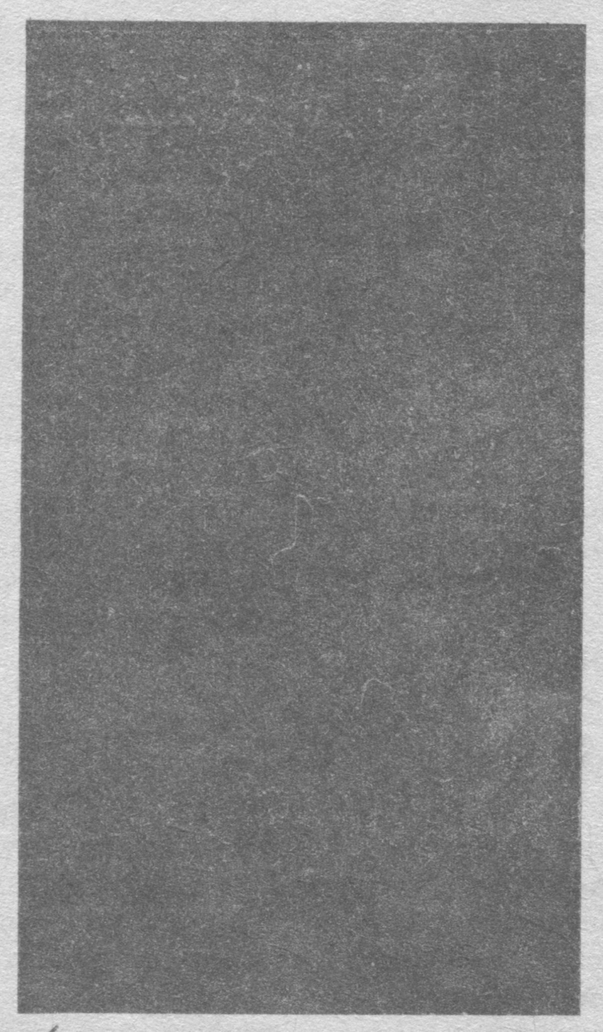
Удаленное выполнение и печать

- RSH позволяет пользователю выполнять одиночные команды на удаленной ЭВМ без использования пароля.
- REXEC позволяет пользователю выполнять одиночную команду на удаленной ЭВМ, предварительно сообщив пароль.
- RPR позволяет пользователю отправить задание на печать на удаленный принтер.
- RPS позволяет пользователю проверить состояние задания в очереди на удаленном принтере.
- RPD позволяет пользователю удалить задание из очереди на удаленном принтере.

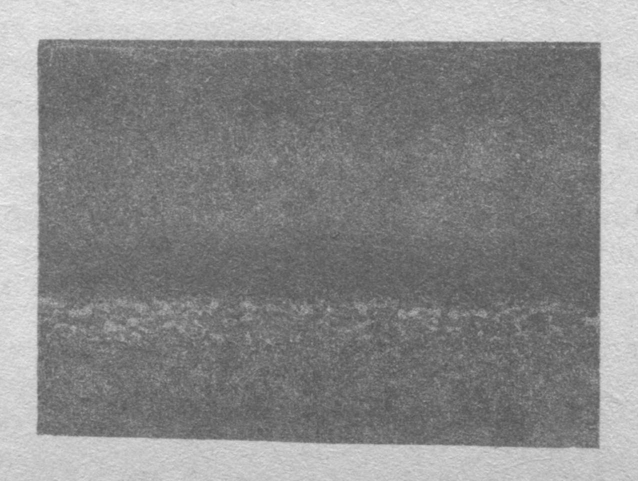
Транспортная система TCP/IP

- Предоставляет набор протоколов TCP/IP, средство сетевого управления программным обеспечением TCP/IP и драйверами NDIS для адаптеров Novell EXOS, 3Com EtherLink и IBM Token-Ring.

Не забудьте подписаться на единственный (по-прежнему) ежемесячный компьютерный журнал. В следующем году он доставит Вам много приятных часов.







Интерфейс прикладного программирования NetBIOS

- RFC 1001 и 1002, удовлетворяющие реализации B-node NetBIOS, позволяют разработчикам создавать адаптационные прикладные программы, использующие интерфейс этой реализации NetBIOS для доступа к TCP/IP. NetBIOS-совместимые прикладные программы могут выполняться без модификации. Поддерживаются также прикладные программы клиент-сервера SMB, такие как LAN Manager.

Поддерживаемые стандарты

Уровень	Протокол	Стандарт
Приложений/		
Представления	TELNET и опции	RFC 854 MIL-STD 1782
	FTP, TFTP	RFC 959 MIL-STD 1780
Сеанса	Domain Name Service	RFC 1034, RFC 1035
	NetBIOS	RFC 1001, RFC 1002
Транспортный	TCP и опции	RFC 793 MIL-STD 1778
	UDP	RFC 768
Сетевой	IP и опции	RFC 791 MIL-STD 1777
	ICMP	RFC 792
	ARP	RFC 826
	Subnet addressing	RFC 950

Информация для заказа

Продукт	Номер изделия
LAN WorkPlace for OS/2	9870111-00
LAN WorkPlace for OS/2 Developer´s Toolkit (лицензия на 10 пользователей)	9870110-00

LAN WorkPlace for Macintosh

LAN WorkPlace for Macintosh обеспечивает пользователям Apple Macintosh доступ к широкому спектру центральных систем, серверов и рабочих групп ЛВС. Применяя привычный интерфейс Macintosh, пользователи осуществляют быстрый и простой доступ к ресурсам сети TCP/IP, включающей персональные компьютеры IBM, миниЭВМ VAX, большие ЭВМ IBM, рабочие станции UNIX и периферийные устройства, подключенные к центральным системам, такие как принтеры и плоттеры. Удаленный сеанс по своим внешним признакам не отличается от работы на Macintosh.

LAN WorkPlace for Macintosh состоит из системного транспортного ПО TCPort, сетевой прикладной программы HostAccess и рабочих средств NetStat. HostAccess предоставляет пользователям набор мощных, придерживающихся промышленных стандартов утилит, обеспечивающих возможность эмуляции терминала и передачи файлов.

Характеристики

- Возможность одновременного использования программ TCP/IP и NetWare for Macintosh.
- Одновременная поддержка сеансов с одной или несколькими центральными системами в сети TCP/IP.
- Поддержка иностранных алфавитов, позволяющая переводить меню и интерфейсы пользователя на множество европейских языков.
- Эмуляция терминала VT100, при которой возможна распечатка частей сеанса TELNET, таких как сообщения электронной почты, непосредственно с экрана.
- Поддержка протокола File Transfer Protocol (FTP) позволяет пользователям просматривать локальные и удаленные директории, осуществлять обмен файлами между системами Macintosh и центральными системами, а также между центральными системами, пользуясь интерфейсом мыши.
- Утилита WHOIS/FINGER позволяет устанавливать связь с другими пользователями, просматривая и выбирая их адреса в электронной почте или из протокола последнего подключения к центральной системе.
- Поддержка программы MultiFinder фирмы Apple Computer позволяет пользователям работать с одной прикладной программой, в то время как другая прикладная программа выполняется в качестве фоновой.
- Транспортные драйверы TCP/IP позволяют использовать прикладные программы других поставщиков, написанные для TCPort Developer´s Toolkit (включая серверы X-Window и SQL-программы клиента).
- Рабочие средства NetStat предназначены для вывода сетевой статистики и предоставляют эффективный инструмент для разработки и диагностики.

Реализация

LAN WorkPlace for Macintosh предназначен для использования в неоднородной вычислительной среде, в которой пользователям Macintosh требуется осуществлять доступ к вычислительным ресурсам всей организации, начиная от больших и миниЭВМ и кончая персональными компьютерами и инженерными рабочими станциями, по стандартному протоколу TCP/IP. При использовании LAN WorkPlace for Macintosh вместе с NetWare for Macintosh обеспечивается одновременный доступ пользователей Macintosh к серверам NetWare и центральным системам TCP/IP, а также обмен файлами, доступ к прикладным программам и связь с пользователями терминалов центральных систем. Кроме того, обеспечивается совокупность распределенных прикладных программ TCP/IP, таких как клиент SQL и сервер X-Window.

Рабочие станции Macintosh, оснащенные LAN WorkPlace for Macintosh, могут быть напрямую связаны с сетями TCP/IP и NetWare через сети LocalTalk или EtherTalk. ПЭВМ Macintosh Plus может использовать

имеющееся в ней соединение с LocalTalk для обеспечения сетевой связи. ПЭВМ Macintosh SE, SE30, II, IIcx и IIci могут использовать как LocalTalk, так и EtherTalk. Продукт Shiva FastPath, который применяется для связи сетей LocalTalk и Ethernet, также может использоваться с LAN WorkPlace for Macintosh.

LAN WorkPlace for Macintosh прост в установке и использовании. Применяя средства конфигурации, входящие в драйвер LAN WorkPlace TCP/IP, супервизор системы или пользователь может в считанные минуты установить программное обеспечение, необходимое для начала работы в сети TCP/IP.

Документация по LAN WorkPlace for Macintosh содержит приложение, в котором описаны различные способы модификации продукта при помощи редактора для поддержки международных таблиц перевода символов.

Разработчики программного обеспечения могут дополнительно приобрести инструментальный набор TC-Port Developer´s Toolkit, который обеспечивает системное транспортное ПО, и интерфейс прикладного программирования в виде библиотеки программных гнезд.

Необходимое аппаратное обеспечение

LAN WorkPlace for Macintosh требует использования Macintosh SE, SE/30 или Plus или любого представителя семейства ПЭВМ Macintosh II с объемом ОЗУ 1 Мбайт и объемом жесткого диска минимум 800 Кбайт. Macintosh может быть подсоединен к сети Ethernet одним из следующих способов: либо он должен находиться в сети LocalTalk, соединенной с сетью Ethernet шлюзом (например, Shiva FastPath 4), либо он должен быть напрямую соединен с сетью Ethernet через сетевой адаптер Ethernet.

Необходимое программное обеспечение

LAN WorkPlace for Macintosh предназначен для работы в операционной системе Macintosh версии 6.02 и выше. Для обеспечения доступа к серверам NetWare, LAN WorkPlace for Macintosh требует использования ПО AppleShare, поставляемого вместе с ОС Macintosh, операционной системы NetWare версии 2.15 и выше и NetWare for Macintosh.

Спецификации

Средства доступа к центральным системам

FTP (RFC 765)
TELNET (RFC 764)
WHOIS/FINGER

Транспортные и сетевые средства TCP/IP

TCP (RFC 793)	IP (RFC 791, RFC 950)
ICMP (RFC 792)	ARP (RFC 826)
UDP (RFC 768)	Domain Name Service
RARP (RFC 903)	(RFC 1034, RFC 1035)

Информация для заказа

Продукт	Номер изделия
LAN WorkPlace for Macintosh	9870074-00

СЕТЕВЫЕ АДАПТЕРЫ И ДРАЙВЕРЫ АППАРАТНЫХ СРЕДСТВ

NE3200

Novell NE3200 — 32-разрядный адаптер шины сети Ethernet, разработанный по спецификации стандарта Extended Industry Standart Architecture (EISA). Этот новый адаптер разработан специально для поддержки NetWare v3.x, 32-разрядной операционной системы фирмы Novell, а также нового поколения мощных серверов с системной шиной EISA.

Еще недавно архитектура шины часто была фактором, сдерживающим производительность сети. Использование NE3200 в сетях, включающих мощный сервер EISA и NetWare v3.x, обеспечивает высокий уровень производительности, необходимый в больших сетях. NE3200 — это первый сетевой адаптер, использующий преимущества новой архитектуры EISA.

Используя встроенную интерфейсную БИС Intel Bus Master Interface Chip (BMIC) и процессор 80186 с тактовой частотой 10 МГц, NE3200 может производить операции обмена между основной памятью и памятью адаптера напрямую со скоростью 33 Мбайт/с. При этом не требуется обслуживание центрального процессора. Это существенно сокращает количество циклов обработки ЦП, обеспечивает повышение эффективности и общей производительности сети. Поскольку NE3200 передает данные по 32-разрядному информационному каналу, сокращается время на передачу по шине, увеличивается операционное время.

Разработанный для высокоскоростных серверов и мостов EISA, адаптер NE3200 полностью согласован со стандартом ANSI/IEEE 802.3, что обеспечивает совместимость с сетевыми продуктами Novell и других поставщиков.

Обеспечивается соединение по тонкому и толстому кабелям Ethernet. NE3200 оснащен стандартным разъемом BNC и встроенным приемопередатчиком для тонкого кабеля Ethernet. Для связи с толстым кабелем Ethernet, витой парой или оптоволоконным кабелем через внешний приемопередатчик NE3200 имеет 15-штырьковый разъем D-shell.

У NE3200 есть множество программно выбираемых конфигураций, позволяющих устанавливать на сервере или мосту большое количество плат.

Характеристики

- Полная поддержка работы с 32-разрядной шиной EISA.
- Использование преимуществ новой архитектуры EISA для обеспечения высокой надежности и производительности.
- Использование встроенной интерфейсной ИС Intel Bus Master Interface Chip (BMIC) и процессора 80186 с тактовой частотой 10 МГц позволяет разгрузить центральный процессор сервера.
- Полное соответствие стандарту ANSI/IEEE 802.3, что обеспечивает совместимость с сетевыми продуктами Novell и других поставщиков.
- Обеспечение прямого соединения с толстым и тонким кабелем Ethernet соответственно через стандартный 15-штырьковый D-разъем и разъем BNC.
- Обеспечение более простого конфигурирования системы, поскольку прерывание, тип прерывания и тип разъема Ethernet могут выбираться программно.

Необходимое программное обеспечение

NE3200 — это адаптер сервера ЛВС NetWare v3.x, требующий использования программных драйверов NetWare для обеспечения связи по сети. Программные драйверы для этого адаптера поставляются с операционной системой NetWare 386 и с самим адаптером. Использование этого адаптера на сервере NetWare для 286 процессора и на рабочей станции EISA в настоящее время не поддерживается. Как только появятся драйверы для таких применений NE3200, об этом будет сообщено через информационное бюро NetWire фирмы Novell.

Спецификации

Поддерживаемые системы
32-разрядные серверы EISA
Шинный интерфейс
32-разрядная шина EISA
Возможные прерывания
Прерывания 5, 9, 10, 11 и 15, выбираемые программно при инсталляции
Адаптер ID: NVL0701

Информация для заказа

Продукт	Номер изделия
@TAB = NE3200 32-bit EISA adapter	883-000815-001
20-Foot RG-58 cable w/T-connector	883-000131-020
Ethernet Terminator Kit	883-000140-001

Адаптеры, изготавливаемые по лицензии Novell

Novell продала лицензии на следующие продукты другим поставщикам. Для получения информации для заказа следует обращаться к этим поставщикам.

Продукт	Поставщик	Телефон
Интеллектуальные контроллеры Ethernet EXOS		
EXOS 201, M2 (NCR)	Federal Technology Corp.	703-739-0500
EXOS 201, Multibus,128K	Federal Technology Corp.	703-739-0500
EXOS 201, Multibus,256K	Federal Technology Corp.	703-739-0500
EXOS 203, QBUS, 512K	Federal Technology Corp.	703-739-0500
EXOS 204, UNIBUS,512K	Federal Technology Corp.	703-739-0500
EXOS 301, Multibus,512K	Federal Technology Corp.	703-739-0500
EXOS 304, UNIBUS, 512K	Federal Technology Corp.	703-739-0500
EXOS 302, VME, option 04	Federal Technology Corp.	739-0500
EXOS 302, VME, option 05	Federal Technology Corp.	739-0500
Адаптеры Ethernet		
NE1000, 8-bit	Eagle Technology	
NE2000, 16-bit	Eagle Technology	
NE/2, 16-bit Micro Channel	Eagle Technology	
NE/2-32, 32-bit Micro Channel	Eagle Technology	
Телефон		
Eagle Technology:	408-441-7453	
Адаптеры Ethernet для Macintosh*		
EtherPort SE v2.0 Rev B	Shiva Corp.	617-252-6300
EtherPort SEL v2.0	Shiva Corp.	617-252-6300
EtherPort II v2.3 Rev H	Shiva Corp.	617-252-6300
EtherPort IIL v2.3 Rev J	Shiva Corp.	617-252-6300
EtherPort SE/30	Shiva Corp.	617-252-6300
EtherPort SE/30L	Shiva Corp.	617-252-6300

* Novell также продала лицензию на технологию EtherPort фирмам Dayna Communications (801-531-0203) и Farallon Computing (415-596-9000). Эти поставщики продают адаптеры Ethernet для Macintosh под другими наименованиями.

NetWare Supplemental Driver Kit (дополнительный набор драйверов)

Novell поставляет набор драйверов локальных сетей NetWare и дисков, которые разработаны другими изготовителями и не входят в операционную систему NetWare. Эти драйверы были протестированы и одобрены испытательной лабораторией фирмы Novell, работающей в рамках программы Independent Manufacturer Support Program.

NetWare Token-Ring Source Routing Drivers (драйверы NetWare, поддерживающие маршрутизацию от источника)

Сетевые мосты соединяют локальные вычислительные сети друг с другом, создавая объединенные сети. Службы маршрутизации позволяют рабочим

группам совместно использовать ресурсы этой объединенной сети. В сети NetWare фирма Novell обеспечивает поддержку различных сред передачи данных, в том числе Token-Ring фирмы IBM. IBM также обеспечивает службы мостов и маршрутизации Token-Ring, реализованные в продукте Token-Ring Network Bridge этой фирмы.

Хотя NetWare Router и IBM Token-Ring Network Bridge обеспечивают одинаковые функциональные возможности, каждый из этих продуктов использует собственную систему маршрутизации данных. NetWare использует метод, называющийся адаптивной маршрутизацией, а IBM использует метод, называющийся маршрутизацией от источника. Вследствие этих различий мосты IBM Token-Ring Network Bridge не могли пропускать трафик NetWare; аналогично, мосты NetWare не могли пропускать трафик IBM.

Эти различия систем маршрутизации очень мало сказывались на пользователях NetWare до тех пор, пока IBM не разработала несколько прикладных программ, написанных с использованием маршрутизации от источника. Пользователи, применяющие прямое соединение Token-Ring для подключения к центральным системам IBM, могут захотеть воспользоваться программой IBM PC 3270 Emulation. Для работы в многокольцевой среде этот продукт требует возможности маршрутизации от источника, имеющейся в IBM Token-Ring Bridge. Если в качестве сетевой операционной системы используется NetWare, применение мостов IBM приводит к изоляции локальных вычислительных сетей друг от друга, поскольку IBM Token-Ring Bridge не может пропускать пакеты NetWare.

Для поддержки таких критических приложений Novell обеспечивает маршрутизацию от источника для рабочих станций DOS и OS/2, а также для серверов NetWare на основе 286 и 386 процессора путем поставки драйверов для сети IBM Token-Ring Network. Эти драйверы позволяют пользователям NetWare осуществлять связь через IBM Token-Ring Bridge. Они также обеспечивают работу прикладных программ IBM, использующих маршрутизацию от источника, в среде NetWare без необходимости их модификации.

Характеристики

- Обеспечивает поддержку маршрутизации от источника в сетях NetWare.
- Драйверы поддерживают адаптеры IBM Token-Ring с пропускной способностью 16 Мбит/с.

Реализация

Реализуя NetWare в среде Token-Ring, супервизоры сети могут:
- использовать мосты NetWare со стандартными драйверами NetWare;

- использовать компоненты обеспечения маршрутизации от источника для связи с мостами IBM.

Выбор реализации зависит от используемых в сети NetWare/Token-Ring прикладных программ.

Мосты NetWare должны использоваться тогда, когда все прикладные программы сети используют для связи поддерживаемые NetWare средства взаимодействия процессов — SPX/IPX, NetBIOS или Named Pipes. Стандартные мосты NetWare могут осуществлять маршрутизацию трафика, созданного любыми прикладными программами, написанными для этих интерфейсов, по сетям с многокольцевой структурой, включая прикладные программы сервера и распределенные прикладные программы. Эти службы маршрутизации делают доступными пользователям любого из подсоединенных колец такие прикладные программы, как серверы баз данных фирм Novell, Gupta Technologies и Oracle.

Использование мостов NetWare также обеспечивает возможность интеграции других сетевых сред, таких как Ethernet, с сетями Token-Ring. Мосты NetWare обеспечивают "прозрачную" маршрутизацию трафика через неоднородные среды, делая одни и те же прикладные программы доступными пользователям любого подсоединенного участка сети.

Примерами программных продуктов, требующих использования драйверов маршрутизации от источника, могут быть NetBIOS (NetBEUI) фирмы IBM, Database Manager и Communications Manager для OS/2 Extended Edition 1.3 фирмы IBM, а также NetWare 3270 Token-Ring Multi Workstation фирмы Novell. Поскольку эти продукты требуют маршрутизацию от источника, они могут использоваться только с IBM Token-Ring Network Bridge. Драйверы Source Routing Drivers и NLM фирмы Novell также должны использоваться для поддержки программ управления сетью фирмы IBM в сетях NetWare.

Мосты IBM и NetWare могут одновременно использоваться в одной сети, но создание таких сетей требует от супервизора тщательного планирования. Прикладные программы, требующие маршрутизации от источника, могут выполняться только через мосты IBM Token-Ring Network Bridge, что накладывает на пользователей таких программ необходимое условие — находиться в сети Token-Ring, использующей драйверы маршрутизации от источника (Source Routing Drivers). Пользователи, которые не связаны с прикладными программами, использующими маршрутизацию от источника, могут находиться в других сетях, таких как Ethernet, соединенных с сетью Token-Ring через мост NetWare Router.

Необходимое аппаратное обеспечение

NetWare Router требует использования сетевого адаптера для каждого участка сети, связанного с мостом. Этот мост может быть реализован как на сервере NetWare (не требуя дополнительного аппаратного обеспечения), так и на внешнем мосту (тогда какой-либо узел сети должен работать в качестве моста).

IBM Token-Ring Bridge требует использования выделенной ПЭВМ, совместимой с IBM PC AT или PS/2, а также двух специальных адаптеров IBM Token-Ring, поставляемые фирмой IBM.

Необходимое программное обеспечение

Программное обеспечение моста NetWare Router поставляется вместе с NetWare v2.2 и NetWare v3.11. Использование драйверов маршрутизации от источника необходимо на каждом сервере или рабочей станции, требующей этого метода маршрутизации. Эти драйверы предназначены для использования на сервере, работающем в операционной системе NetWare версии 2.1 и выше, и на рабочих станциях, работающих в ОС DOS версии 3.1 и выше или OS/2 версии 1.2 и выше.

Мост IBM Token-Ring Network Bridge требует использования программного обеспечения фирмы IBM, которое может быть куплено у этой фирмы или через ее уполномоченного торгового агента.

Remote Boot PROMS (ПЗУ удаленной загрузки)

Многие сетевые адаптеры оснащены гнездами для ПЗУ (PROMS) удаленной загрузки. Эти ПЗУ позволяют пользователям загружать с сервера операционную систему DOS и необходимое программное обеспечение сетевой связи на бездисковые рабочие станции.

Novell поставляет ПЗУ удаленной загрузки для всех своих сетевых адаптеров, что позволяет использовать их на бездисковых рабочих станциях. Novell также поставляет ПЗУ удаленной загрузки для широкого спектра сетевых адаптеров других изготовителей, что также позволяет их использовать на рабочих станциях без дисководов в сети NetWare. Такие ПЗУ, предназначенные для использования с операционными системами NetWare версий 2.1 и выше, имеются для адаптеров: 3Com EtherLink, Corvus Omninet, Standard Microsystems или Pure Data ARCNET, а также Micom-Interlan 5010 и Novell S-Net INIC.

Мы продолжаем публикацию фрагментов из книги Д.Рогаткина и А.Федорова “Turbo Pascal в среде Windows“. Этот раздел посвящен описанию библиотеки ObjectWindows, разработанной корпорацией Borland International для программирования в среде Microsoft Windows.

TURBO PASCAL в среде WINDOWS

Введение в библиотеку ObjectWindows

Программирование в среде Windows — отнюдь не простая задача, поскольку программа, во-первых, должна иметь управляемую событиями архитектуру, не привычную для традиционного программирования, а во-вторых, грамотно использовать более 600 функций и 20´ сообщений ядра Windows, назначение и применение которых программисту желательно знать. Упростить программирование позволяет использование объектно-ориентированных библиотек, к числу которых относится и ObjectWindows, разработанная корпорацией Borland.

Одна из особенностей объектно-ориентированных расширений языков корпорации Borland — использование динамических виртуальных методов — позволила отказаться от громоздкого оператора выбора, появляющегося в теле программы обработчика событий (здесь — оконной функции).

К числу основных достоинств библиотеки ObjectWindows относятся:
- последовательный, интуитивно понятный и простой в использовании интерфейс с ядром Windows, что дает возможность создавать приложения, работающие под Windows, даже программистам, ранее не работавшим в этой среде;
- автоматическая обработка сообщений и автоматическое управление окнами, минимизирующие трудоемкость создания программы;
- возможность повторного использования и расширяемость кода, также позволяющие снижать трудоемкость программирования.

ООП и ObjectWindows

Известны три принципа ООП, каждый из которых способствует одновременному решению нескольких задач.

1. **Инкапсуляция** — объединение данных и процедур, позволяющее создавать объекты, соответствующие интерфейсным элементам Windows. Набор методов такого объекта полностью контролирует поведение соответствующего элемента интерфейса.
2. **Наследование** — позволяет создавать объекты, соответствующие интерфейсным элементам с заданными свойствами, а также дополнять такие объекты набором методов обработчика сообщений. Таким образом, наследование дает возможность снижать объемы программирования при создании объектов.
3. **Полиморфизм** — обеспечивает единую линию поведения различных объектов при вызове одного и того же метода. Это особенно важно в связи с тем, что приложение в среде Windows представляет собой иерархическую систему окон с различными свойствами, а множество методов использует итераторы для выполнения однотипных действий над различными дочерними объектами. Так, при выполнении метода **Create** для всех дочерних объектов осуществляется создание интерфейсного элемента для соответствующего объекта, будь то окно, элемент управления или панель диалога. Полиморфизм также дает возможность программисту сократить число используемых методов.

Наличие виртуальных методов усиливает принцип наследования, позволяя экономить на количестве переопределяемых методов.

Иерархия объектов

Иерархия объектов ObjectWindows отражает наличие общих интерфейсных элементов Windows-приложений, — окон, панелей диалога, элементов управления, а абстрактный объект **TAplication** описывает шаблон приложения в идеологии Windows. Дерево объектов ObjectWindows представлено на рисунке.

Большинство объектов библиотеки ObjectWindows относятся к интерфейсу, а их экземпляры соответ-

ствуют реальным интерфейсным элементам. Корнем ветви интерфейсных объектов является абстрактный объект **TWindowsObject**, который определяет общие черты поведения всех своих потомков*. Производными от **TWindowsObject** являются объекты **TWindow**, **TDialog** и **TControl**, которые выступают в качестве родоначальников основных групп интерфейсных элементов — окон, панелей диалога и элементов управления. **TWindowsObject** определяет основной набор методов для всех элементов, наследующих его свойства. Каждый конкретный тип объекта переопределяет и дополняет этот набор, тем самым обеспечивая уникальность своего поведения.

Библиотека также содержит объект **TApplication**, экземпляр которого выполняет все действия приложения, включая инициализацию, обработку сообщений, завершение приложения и разрешение ошибочных состояний. Производный от **TApplication** объект инициализирует главное окно с конкретным поведением. Возможно также переопределение ряда методов, если требуется создать приложение со специфическим поведением.

Стандартные инструментальные объекты выполняют распространенные специализированные задачи. Это объекты **TEditWindow** и **TFileWindow**, предназначенные для создания редакторов текста, **TFileDialog** и **TInputDialog**, предназначенные для ввода имени файла и текстовой строки. Объекты представляют собой законченный инструментарий, но могут наращивать и видоизменять свои функции за счет создания потомков.

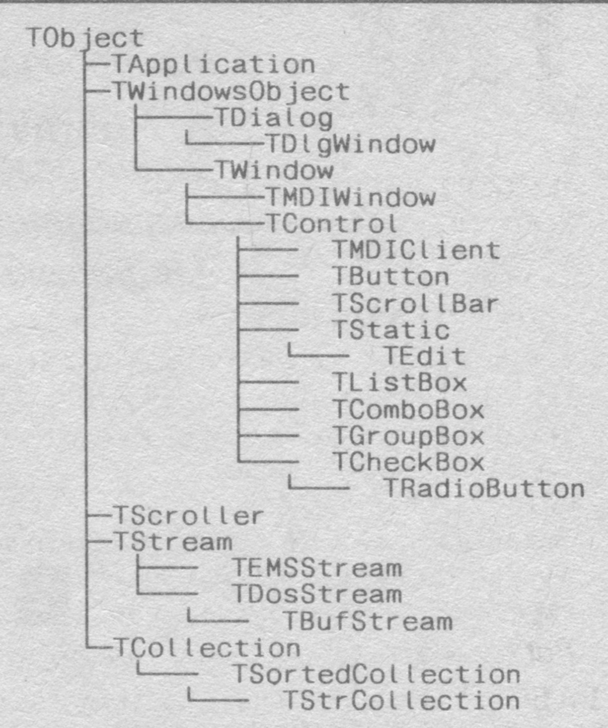
Библиотека содержит набор дополнительных объектов общего назначения, используемых в некоторых операциях и самими интерфейсными объектами, — это TCollection и TStream.

Абстрактные объекты ObjectWindows

Для понимания устройства и механизма работы всех объектов библиотеки полезно рассмотреть абстрактные объекты, стоящие во главе основных ветвей. Подробно об объектах, экземпляры которых могут соответствовать реальным элементам Windows, будет рассказано в соответствующих частях книги.

TApplication

Для того чтобы создать конкретное приложение под Windows с помощью ObjectWindows, необходимо создать объект — потомок **TApplication**. Чтобы понять, какие методы и каким образом необходимо переопределить в объекте-потомке для создания приложения с заданными свойствами, необходимо



рассмотреть функции, выполняемые методами TApplication:
- инициализация данного экземпляра приложения;
- инициализация первого экземпляра приложения;
- создание и отображение главного окна;
- обеспечение обработки сообщений;
- завершение приложения.

Отсюда выделяется ряд методов, осуществляющих определенную работу. Какие же методы использует конструктор для выполнения своей работы?

Конструктор инициализирует ряд полей — Name, MainWindow, Status, HAccTable. Для первого экземпляра приложения вызывается метод **InitApplication**. Для каждого экземпляра приложения вызывается метод **InitInstance**, который в свою очередь вызывает метод **InitMainWindow**, который инициализирует и создает (посредством метода **MakeWindow**) главное окно, а затем и отображает его на экране. Метод **Run** запускает цикл обработки сообщений — **MessageLoop**. Для обеспечения специальной обработки сообщений можно переопределить метод **ProcessAppMsg**, к которому обращается **MessageLoop**. **ProcessAppMsg** выполняет трансляцию сообщений для немодальных диалогов, акселераторов и акселераторов MDI посредством методов **ProcessDlgMsg**, **ProcessAccel** и **ProcessMDIAccel** соответственно. В зависимости от особенностей приложения можно ускорить трансляцию сообщений, упростив один из перечисленных методов или вообще исключив его, как это сделано в приведенном ниже примере.

Придание приложению необходимых свойств

Для обеспечения различного поведения первого и последующих экземпляров приложений необходимо переопределить методы **InitApplication** и **InitInstance**. Поскольку **InitApplication** вызывается только первым экземпляром приложения, то переопределив этот ме-

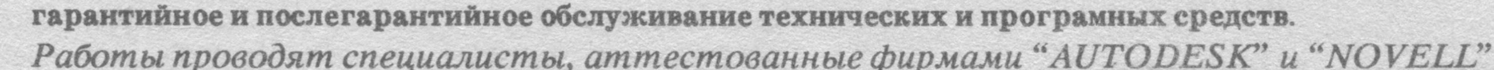
* Во избежание путаницы, для обозначения родственных отношений между объектами будут использованы термины "предок-потомок", а для родственных отношений между окнами — "родитель-ребенок".



АССОЦИАЦИЯ МАЛЫХ НАУЧНО-ПРОИЗВОДСТВЕННЫХ ПРЕДПРИЯТИЙ
“Т Е М А”

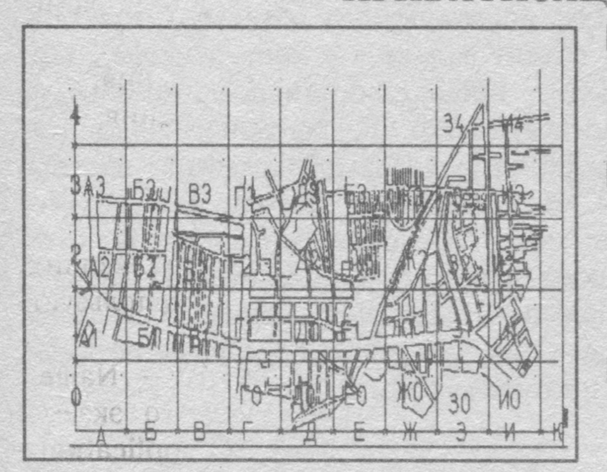
Официальный дилер фирм “AUTODESK” и “NOVELL”

- ☐ **ПАКЕТ ПРОГРАММ AUTOCAD**
- ☐ **услуги по созданию локальных сетей**
  - • Novell NetWae v3.11, v2.2, v2.15;
  - • Novell NetWare Lite;
  - • D-Link LANsmart 3.0, v2.02;
- ☐ **программное обеспечение фирмы “NOVELL”**
- ☐ **сетевое оборудование Arcnet, Ethernet**
- ☐ **компьютеры AT-386/33 (гарантируется работа в качестве файл-сервера)**
- ☐ **модемы 2400 + MNP5**
- ☐ **факс-платы FAX-96 (США), адаптированные к отечественной телефонной сети**
- ☐ **гарантийное и послегарантийное обслуживание технических и програмных средств.**



Работы проводят специалисты, аттестованные фирмами “AUTODESK” и “NOVELL”

РАБОТНИКАМ МУНИЦИПАЛЬНЫХ СЛУЖБ, ПРЕФЕКТУР И МЭРИЙ, ОРГАНАМ ПО ПРИВАТИЗАЦИИ И ТЕХНИЧЕСКОЙ ИНВЕНТАРИЗАЦИИ



☐ **АРМ “БЮРО ТЕХНИЧЕСКОЙ ИНВЕНТАРИЗАЦИИ”**

— связь обьектов с графическими файлами поэтажных планов помещений и соответствующими записями баз данных: технических паспортов помещений, владельцев помещений, арендаторов.

☐ **ИНФОРМАЦИОННО-СПРАВОЧНЫЕ СИСТЕМЫ, СВЯЗАННЫЕ С КАРТОГРАФИЕЙ**

— карты района, обьекты и поэтажные планы.

СЛУЖБАМ, СВЯЗАННЫМ С ЭКСПЛУАТАЦИЕЙ ДОРОГ, ТЕЛЕФОННЫХ, ГАЗОВЫХ И ДРУГИХ КОММУНИКАЦИЙ

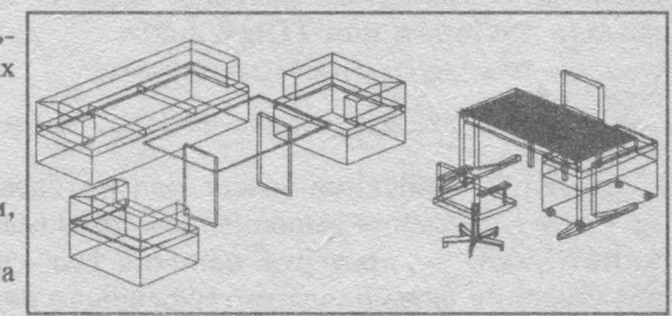
☐ **АРМ “ИНЖЕНЕРНЫЕ СЕТИ”**

— информация о технических характеристиках инженерных сетей, эксплуатационных данных, владельцах, — непосредственно из карты района и из текстовой базы данных.

ПРОЕКТНЫМ, ПРОЕКТНО-СТРОИТЕЛЬНЫМ И МОНТАЖНЫМ ОРГАНИЗАЦИЯМ

☐ **АРМ ГРАФМАКЕТ**

— автоматизированная компоновка механизмов и вспомогательных объектов при строительных, горнодобывающих, дорожных и других работах;



☐ **АРМ “КОНСТРУКТОР”**

— автоматический выпуск текстовой документации по ЕСКД.

— работа с развитой графической базой данных по стандартным, покупным изделиям и материалам.

— автоматическое формирование и поддержка структуры проекта на изделие

РАЗРАБОТЧИКАМ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

☐ **БИБЛИОТЕКИ СХЕМНЫХ И КОНСТРУКТОРСКИХ ЭЛЕМЕНТОВ для СИСТЕМЫ PCAD**

— 10500 элементов, 320 страниц каталога, информационно-поисковая система.

☐ **Универсальный транслятор PCAD – AUTOCAD,**

— масштаб, зеркало, поворот, перевод в ACAD в миллиметрах или дюймах, штриховка полигонов под любым углом, заливка полигонов, вывод окружностей без сегментации.

ПРЕДПРИЯТИЯМ И ОРГАНИЗАЦИЯМ ЛЮБОЙ ФОРМЫ СОБСТВЕННОСТИ

☐ **АРМ “Юрист”**

— хранение текстов юридических документов (законов, постановлений, методик, инструкций и т.п.)

☐ **АРМы Склад, Бухгалтерия.**

Контактные телефоны (095) 118-6356, 118-6347 (с 9 до 18),
факс: (095) 118-6638 (круглосуточно на автомате)
115487, г. Москва, ул.Садовники, 4, МНПП “ТЕМА”

тод, можно обеспечить, например, загрузку таблицы акселераторов или предусмотреть выполнение только одного экземпляра приложения.

```
Uses WinTypes, WinProcs, WObjects;
Type
Tsinglapplication = Object(TApplication)
firstapp : boolean;
procedure InitApplication; Virtual;
procedure InitMainWindow; Virtual;
function ProcessAppMsg(var Message: TMsg): Boolean; Virtual;
end;
procedure Tsinglapplication.InitMainWindow;
begin
   if firstapp then
     mainWindow := New(PWindow,
                     Init(Nil, 'Single Application'))
   else
   begin
     MessageBox(0, 'Только один экземпляр может быть запущен',
          'Single Application', MB_OK or mb_ICONINFORMATION);
     Halt(1)
   end;
  end;
procedure Tsinglapplication.InitApplication;
  begin
    firstapp := True
  end;
function Tsinglapplication.ProcessAppMsg(var Message: TMsg): Boolean;

begin
  ProcessAppMsg := False
end;

Var
ExampApp: Tsinglapplication;

begin
   ExampApp.Init('ExampApplication');
   ExampApp.Run;
   ExampApp.Done
end.
```

Для создания главного окна с определенными свойствами переопределяется метод **InitMainWindow**, в котором инициализируется главное окно и указатель на него присваивается полю MainWindow. Инициализируемое главное окно, как правило, является наследником объекта **TWindow** или **TDlgWindow**.

```
procedure TExampApplication.InitMainWindow;
Begin
  MainWindow := New(PExampWindow, Init(Nil, 'Params'))
end;
```

Объект **TApplication** имеет ряд методов, необходимых для управления выполнением приложения:

ValidWindow — проверка наличия свободной памяти для создания экземпляра интерфейсного элемента;

MakeWindow — создание интерфейсного элемента, соответствующего экземпляру объекта, посредством метода объекта **Create**, с предварительной проверкой **ValidWindow**;

ExecDialog — выполнение модальной панели диалога;

Error — обработка ошибочной ситуации с выдачей панели сообщения;

CanClose — проверка возможности закрытия приложения путем вызова соответствующего метода для главного окна.

TWindowsObject

Основой всех объектов ObjectWindows, соответствующих интерфейсным элементам Windows, является абстрактный объект TWindowsObject.

Каждому интерфейсному элементу поставлен в соответствие некоторый объект ObjectWindows. ObjectWindows не является частью ядра Windows, поэтому объекты ObjectWindows не реализуют сами интерфейсные элементы, как это происходит в Turbo Vision. Тем не менее, набор полей и методов объекта ObjectWindows позволяет полностью контролировать поведение реального элемента Windows. Объект содержит основной набор методов для инициализации, построения, отображения, обработки сообщений и уничтожения интерфейсного элемента Windows. Поля объекта содержат атрибуты создания, включая иконку, курсор, местоположение, размер и т. д., информацию о состоянии соответствующего интерфейсного элемента, вспомогательные флажки, указатель на родительский элемент и список дочерних элементов. Каждый оконный объект способен обрабатывать сообщения посредством методов — обработчиков сообщений. Инициализация любого объекта ведет к автоматическому включению указателя на него в список дочерних элементов ChildList родительского объекта (указатель на родительский объект задается в конструкторе). ObjectWindows разделяет процесс инициализации объекта и создания соответствующего интерфейсного элемента. Это позволяет изменить некоторые параметры объекта перед созданием интерфейсного элемента и создать интерфейсный элемент со свойствами, отличными от стандартных. Таким образом, в большинстве случаев отпадает необходимость создания потомков базовых объектов для получения заданного поведения создаваемого интерфейсного элемента.

Итераторы для работы с дочерними элементами

Для выполнения ряда действий над всеми дочерними объектами предназначены методы-**итераторы**. В ObjectWindows включены итераторы **ForEach** и **FirstThat**.

Первый позволяет выполнить некоторое действие над всеми дочерними объектами, например, выключить все кнопки с независимой фиксацией:

```
procedure ExampWindow.UnCheckAllBoxes;
 procedure UnCheckTheBox(ABox: PCheckBox); far;
 begin
  ABox^.UnCheck
 end;
begin
 ForEach(@UnCheckTheBox)
end;
```

Второй выполняет проверку дочерних объектов, пока не будет найден первый, удовлетворяющий заданному условию. Следующий пример иллюстрирует поиск дочернего окна с необходимым ID (таким образом реализован метод ChildWithId):

```
procedure TPaletteWin.DrawItem(var Msg: TMessage);
var
  rc  :  TRect;
  pdi :  PDrawItemStruct;
function IsThisIDEq(AButton: PWindowsObject): Boolean;far;
```

```
begin
   IsThisIDEq:= PControl(AButton)^.Attr.Id = pdi^.CtlID;
end;

begin
  ....
  DrawIcon(pdi^.hDC, rc.left, rc.top,
      HICON(PODButton(FirstThat(@IsThisIDEq))^.hData));
end;
```

Параметром итератора является адрес процедуры, выполняющей операцию над каждым дочерним элементом. Эта процедура должна иметь один параметр, совместимый по типу с указателем. Параметр содержит указатель на обрабатываемый дочерний объект, поэтому в заголовке можно сразу задать тип указателя на обрабатываемый объект, если дочерние объекты однотипны, и исключить применение префикса преобразования типа, что и сделано в первом примере. Следует отметить, что итерируемая процедура должна быть локальной и иметь в заголовке признак **far** (дальний метод вызова).

Создание любого интерфейсного элемента производится вызовом метода **Create**, который, в свою очередь, после создания элемента Windows, вызывает метод **SetupWindow**, создающий дочерние окна из списка ChildList. В методе **SetupWindow** можно предусмотреть действия, необходимые для инициализации вновь созданного интерфейсного элемента.

Когда объект получает сообщение WM_CLOSE, вызывается метод **CloseWindow**, который в свою очередь вызывает метод **CanClose**. Если этот метод возвращает True, то окно закрывается и соответствующий объект освобождает память, в противном случае команда на закрытие игнорируется. Метод **CanClose** вызывает аналогичный метод для всех дочерних окон, идя по списку ChildList. Поэтому для сохранения названной особенности необходимо вызывать **CanClose**-предка при переопределении этого метода.

TWindow

Создание интерфейсного элемента происходит в два этапа. На первом вызывается конструктор **Init** соответствующего объекта и устанавливаются атрибуты интерфейсного элемента. На втором — используется метод объекта TApplication **MakeWindow** для создания интерфейсного элемента.

После создания элемента значение поля **HWindow** становится определенным и содержит ссылку на созданный интерфейсный элемент. Для создания интерфейсного элемента метод **MakeWindow** использует метод **Create**. Этот метод, используя вызов соответствующей функции ядра Windows, создает интерфейсный элемент и подключает к нему оконную функцию, а затем обращается к методу **SetupWindow** для настройки созданного элемента на работу. **SetupWindow** автоматически создает все интерфейсные элементы, находящиеся в списке дочерних объектов, в соответствии с установкой у них флажка разрешения/запрещения автосоздания. Для управления автоматическим созданием дочерних объектов после их инициализации следует использовать методы **DisableAutoCreate** и **EnableAutoCreate**. По умолчанию, окна имеют разрешенный флаг автосоздания, а диалоговые панели — запрещенный. При использовании ObjectWindows созданный интерфейсный элемент Windows не принимает сообщение wm_Create, поэтому действия, соответствующие этому сообщению, надо включить в метод **SetupWindow**. Следует отметить, что переопределяя метод **SetupWindow**, необходимо не забывать обращаться сначала к соответствующему методу предка, для того чтобы были созданы дочерние элементы. Поскольку во время выполнения метода **SetupWindow** поле HWindow определено, то можно выполнить необходимые действия по настройке созданного интерфейсного элемента, например, видоизменить системное меню. Другой распространенной операцией, проводимой в этом методе, является начальная установка дочерних элементов, например, заполнение списков, установка переключателей и т.п. Появится на экране созданный интерфейсный элемент-окно или нет, зависит от установки флага ws_Visible поля Attr.Style, о чем будет рассказано ниже. В дальнейшем управлять появлением и исчезновением созданного элемента можно при помощи метода **Show**. Для уничтожения элемента используется метод **Done**, который в свою очередь выполняет уничтожение всех дочерних элементов. Благодаря этому, нет необходимости явного удаления дочерних элементов при уничтожении окна.

Многие методы объекта **TWindow** взаимодействуют с объектом **TScroller**, который может использоваться для упрощения прокрутки в окне. Объект **TScroller** поддерживает большое виртуальное пространство экрана путем построения зависимости между позицией бегунка полосы прокрутки в окне и 32-битовой виртуальной координатной системой.

TControl

Объект **TControl** соответствует специализированному дочернему окну, предназначенному для осуществления операций ввода-вывода. Этот объект является производным от **TWindow** в иерархии объектов ObjectWindows. Все объекты, соответствующие элементам управления, являются производными от **TControl**. Поведение элементов управления определяется ядром Windows. ObjectWindows несколько расширяет набор функций, выполняемых элементом управления. К числу таких расширений относится возможность автоматического обмена данными с элементами управления.

TDialog

Объект **TDialog** соответствует специализированному типу родительского окна, осуществляющего обмен информацией с пользователем. Чаще всего диалоговые панели создаются (или выполняются) модально на короткий промежуток времени. Существует возможность создания немодальных панелей диалога, во время выполнения которых не блокируется взаимодействие с другими окнами приложения.

Объект **TDialog** — производное от **TWindowsObject** и определяет поведение диалоговой панели.

Объект поддерживает механизм передачи данных от содержащихся в нем элементов управления. При закрытии панели диалога данные автоматически пересылаются в буфер передачи.

TMDIWindow

ObjectWindows позволяет программисту не заботиться о детальном знании правил MDI-интерфейса. Порядок работы с ObjectWindows слегка отличается от традиционной для MDI-приложений. Программист просто создает экземпляр объекта **TMDIWindow** как главное окно MDI-приложения (называемое обрамляющим окном). Объект **TMDIWindow** является производным от объекта **TWindow**. **TMDIWindow** определяет методы манипулирования дочерними окнами MDI и управления клиентным окном MDI.

Клиентное окно — это дочернее окно, выполняющее специальные функции, — оно помогает управлять другими дочерними окнами MDI внутри обрамления. Этому окну соответствует объект **TMDIClient**. Он является потомком объекта **TControl**, так как функционально клиентное окно MDI и элемент управления схожи.

Поведение, требуемое от дочерних окон MDI, описано в объекте **TWindow**. Никаких специальных действий программиста при работе с такими окнами не требуется. Если представитель **TWindow** является дочерним окном в MDI-приложении, то, по умолчанию, поведение этого окна будет соответствовать требованиям ядра Windows к дочерним окнам MDI. Обработка сообщений дочерним окном MDI происходит точно так же, как и обычным окном. Дочернее окно MDI в фокусе способно принимать команды, выбираемые в родительском меню. Дочернее окно MDI пересылает сообщение родительскому окну, если само не имеет методов для его обработки.

TScroller

ObjectWindows имеет механизм перемещения окошка просмотра в логических координатах просматриваемого пространства. Перемещение производится с помощью полос прокрутки окна, если таковые имеются. Вне зависимости от того, есть полоса прокрутки у окна или нет, каждое окно может иметь объект-прокрутчик, который осуществляет непосредственную прокрутку изображения. Прокрутчик оперирует смещением отображаемой области в логических координатах XPos и YPos в координатном пространстве. Эти смещения и размеры отображения X и Y однозначно определяют фрагмент отображения. Метод прокрутчика **IsVisibleRect** позволяет определить, попадает требуемый отрезок логического пространства в фрагмент отображения или нет. В случае необходимости перерисовки используется метод **Paint**.

Атрибуты XRange и YRange определяют координатное пространство. Для обеспечения большей гибкости процесса прокрутки диапазон значений этих атрибутов может достигать 31-битовой величины. Логический атрибут AutoOrg индицирует, будет ли задействован внутренний механизм прокрутки окна. Атрибуты XLine, XPage, YLine и YPage определяют два значения шага прокрутки. Истинное значение логического атрибута TrackMode заставляет обновлять изображение во время перемещения бегунка по полосе прокрутки мышью.

Обработка сообщений

Для обслуживания сообщений, посылаемых интерфейсному элементу, используются методы — обработчики сообщений. Эти методы являются динамическими виртуальными и содержат в заголовке индекс, который соответствует коду обслуживаемого сообщения.

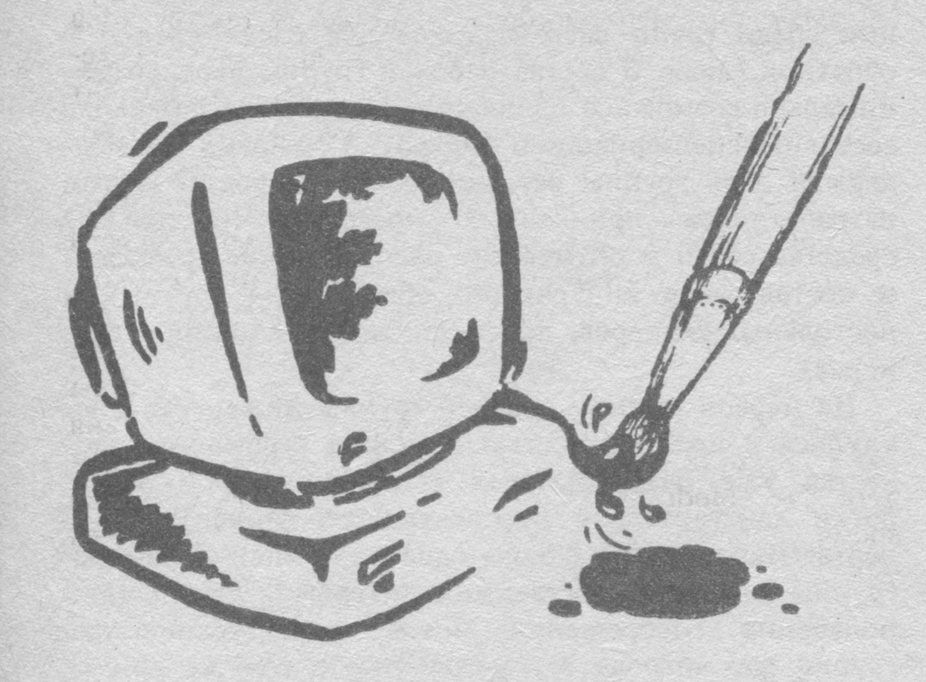
```
procedure RespondMethod(Var Msg: TMessage); virtual RespondIndex;
```

Индекс метода зависит от того, к какой группе принадлежит обрабатываемое сообщение, и выбирается, исходя из правил, приведенных в разделе, посвященном сообщениям. Методу — обработчику сообщения передается один параметр — структура типа TMessage. Эта структура заполняется в соответствии с сообщением Windows и имеет вид:

```
TMessage = record
    Receiver: HWnd;
    Message: Word;
    case Integer of
      0: (
        WParam: Word;
        LParam: Longint;
        Result: Longint);
      1: (
        WParamLo: Byte;
        WParamHi: Byte;
        LParamLo: Word;
        LParamHi: Word;
        ResultLo: Word;
        ResultHi: Word);
  end;
```

Поля структуры соответствуют информации, передаваемой в сообщении. Для удобства обращения с информацией сообщения структура содержит вариантную часть. Если после получения сообщения обработчиком его дальнейшая обработка не требуется, то значение поля Result устанавливается равным единице.

Д.Рогаткин, А.Федоров



Тот, кто хоть раз в жизни собственноручно установил на собственный компьютер сканер, знает, сколько неожиданностей таит в себе этот процесс. Статья, которую передала нам фирма Hewlett-Packard — об интерфейсе TWAIN, грозящем изменить положение с устройствами ввода изображения. Возможно, через пару лет подключение сканера будет не сложнее подключения обычной мыши...

Ввод изображения: головная боль или нормальная работа?

На заре эпохи настольных издательских систем большинство публикаций, создававшихся с их помощью и выводившихся на черно-белый лазерный принтер, содержало только текст и простые графические рисунки. Но время не стоит на месте, и технология становится все более совершенной. Если раньше для создания сложных цветных изданий нужен был электронный цветокорректор — устройство стоимостью порядка полумиллиона долларов, то теперь и профессиональные издатели, и художники могут создавать их при помощи персонального компьютера.

Обычная публикация сегодня может включать черно-белые графические, полутоновые и цветные изображения, получаемые с помощью настольных и ручных сканеров, камер и image capture boards (плат ввода изображения с видеовыхода различных аналоговых устройств), по качеству приближающихся к профессионально подготовленным изданиям.

К сожалению, получение изображения остается достаточно трудоемким и сложным процессом. Для того чтобы получить и разместить изображение в публикации, необходимо выйти из прикладной программы, в которой вы работаете. Затем нужно найти и вызвать драйвер устройства, с которого будет считываться изображение, установить параметры его работы, получить изображение, сохранить его на диске, закрыть драйвер, вернуться в прикладную программу, попасть в то место, где суждено находиться картинке, и, наконец, считать с диска файл, содержащий это изображение. Естественно, этот нудный процесс мало нравится рекламным художникам и издателям, работать таким образом им совершенно неохота, особенно если учесть, что изображения, полученные в реальном масштабе времени, используются все чаще.

Перед поставщиками аппаратного обеспечения встала проблема предоставления клиентам таких изделий, которые не только увеличили бы производительность их труда, но и были простыми в использовании.

Решение проблемы

Когда в 1990 году эта проблема встала в полный рост, разработчики аппаратного и программного обеспечения пытались установить стандарты на базе своих собственных интерфейсов для устройств получения изображения. Хотя, в общем-то, это был шаг в правильном направлении, тиражирование собственных интерфейсов не могло стать решением проблемы, поскольку нельзя требовать от разработчиков прикладных программ писать драйверы для каждого из

поддерживаемых устройств. И тем более не имело смысла умолять поставщиков аппаратного обеспечения писать различные драйверы для обеспечения интерфейса с каждой используемой прикладной программой. Важнее было избавить пользователей от необходимости работы со многими специфичными сочетаниями прикладная программа/драйвер устройства.

Итак, сформулировать проблему можно следующим образом. Пользователям нужен простой безболезненный путь получения данных для применяемых ими прикладных программ. Разработчикам прикладных программ хочется, чтобы их продукты были совместимы с широким набором устройств без необходимости написания и поддержки кучи специальных драйверов. Разработчикам аппаратного обеспечения требуется совместимость с огромным количеством прикладных программ без учета специфики конкретных программ.

Наиболее очевидное решение проблемы в подобной ситуации — это разработать общий открытый промышленный интерфейс для непосредственного получения данных из внешних источников прямо в прикладные программы. В этом случае каждый разработчик программ поддерживает некоторое стандартное средство и метод работы с данными, а каждый поставщик аппаратного обеспечения пишет единственный драйвер для своего устройства. Это означает, что поставщики устройств могут писать драйверы, чтобы представить свои продукты в наиболее выгодном свете, тогда как поставщик прикладных программ освобожден от разработки драйверов или от необходимости поддержки собственного "правильного" интерфейса. Поставщики прикладных программ оказываются в выигрышном положении, потому что один единственный интерфейс будет поддерживать все драйверы устройств, подготовленные любым из поставщиков железа. Поставщикам аппаратного обеспечения необходимо создать всего лишь один стандартный драйвер для своего устройства — затем он может быть использован всеми прикладными программами, поддерживающими стандартные методы работы с данными. Это неплохо еще и потому, что разработчик аппаратуры лучше чем кто-либо еще знает особенности своего детища и может создать наиболее корректный драйвер для него. Пользователи могут, используя этот маленький набор драйверов, обеспечить доступ к своим прикладным программам.

Консорциум TWAIN: история создания и структура

Консорциум TWAIN был организован представителями компаний Aldus, Caere, Eastman Kodak, Hewlett-Packard и Logitech. Целями этой рабочей группы стали: скорейшее разрешение технических проблем относительно создания такого открытого интерфейса, возможно более быстрое распространение его спецификации среди групп разработчиков для ознакомления и обратной связи. В состав рабочей группы входило минимально возможное количество людей (что позволило составить спецификацию быстро), в то время как поддержка этой группы осуществлялась широким кругом разработчиков прикладных программ (настольные средства связи и оптическое распознавание символов) и поставщиков аппаратного обеспечения (ручных и настольных сканеров, профессиональных цветных сканеров).

На первой встрече рабочей группы инженеры столкнулись с необходимостью изучения огромного количества уже разработанных спецификаций и разработки действительно перспективного варианта стандарта. Параллельно с этим одна из компаний рабочей группы написала собственный интерфейс для прямого получения изображения, после чего техническая группа рассмотрела эту разработку с тем же вниманием, что и представленные на обсуждение другими компаниями спецификации, которых оказалось почти тридцать.

Инженеры рабочей группы принимали участие в "лабораторных работах" для дальнейшего определения и уточнения спецификации. Отдельно работала группа по маркетингу, обсуждая вопросы распространения средств разработки интерфейса, вопросы его поддержки и многие другие проблемы.

Цели спецификации TWAIN

Основная цель создания данной спецификации — решить проблему совместимости, легкого объединения различных устройств ввода и генерируемой ими информации с любым программным обеспечением, нуждающимся в получении этой информации. Если говорить более конкретно, то можно выделить следующие цели создания спецификации:

1. **Поддержка различных платформ**. Интерфейс должен работать на Macintosh и в Windows, но в то же время иметь возможность расширения для работы в Unix и других платформах. Использование системно-зависимых механизмов должно быть сведено к минимуму.
2. **Поддержка различных устройств** — включая ручные сканеры, планшетные сканеры, сканеры слайдов, платы ввода видеоизображения и захвата кадров, цифровые камеры и базы данных изображений — весь ряд средств получения изображения.
3. **Возможность работы с различными форматами данных**. Этот основной механизм обмена данными должен позволять передавать данные в существующих стандартных форматах и в собственных форматах файла, таких как TIFF, PICT и DIP. Кроме того, необходимо принимать во внимание полезность обработки данных, не подпадающих под общепринятое понятие компьютерного изображения (таких как текст, данные факса или векторная графика).

4. **Повсеместное использование**. Работать через открытый и стандартный интерфейс выгодно. Поэтому разработчикам аппаратного и программного обеспечения следует снабдить свои устройства необходимыми драйверами или включить соответствующую поддержку в свои прикладные программы.
5. **Расширяемость и совместимость с новыми версиями**. Архитектура спецификации должна расширяться по мере совершенствования технологий и должна иметь возможность дополнения новыми описаниями, неизвестными сегодня. Новые версии спецификации будут совместимы "сверху вниз" с написанными ранее спецификациями.

Архитектура

TWAIN обеспечивает простую, но разнообразную методологию для организации связи соответствующих спецификации TWAIN-прикладных программ с соответствующими TWAIN-устройствами. То, как прикладные программы взаимодействуют с источником исходных данных, может быть описано посредством четырехуровневого протокола: Application Layer (уровень прикладной программы), Protocol Layer (уровень протокола), Acquisiition Layer (уровень получения изображения) и Device Layer (уровень устройства). Каждый этап получения изображения соответствует одному из этих уровней и обслуживается программой, которая связана через Source Manager (менеджер исходных данных) с Source-драйвером (драйвером источника данных), причем последний рассматривается как физическое устройство, генерирующее изображение.

Давайте посмотрим, каким образом происходит процесс получения изображения при использовании TWAIN. В первую очередь пользователь выбирает меню "Acquire" из прикладной программы. "Acquire" применяется в том случае, когда подключено только одно устройство ввода или когда пользователь хочет воспользоваться устройством ввода, которое перед этим применялось. Интерфейс пользователя в прикладной программе содержит также опцию "Select Source". Пользователь обращается к ней при необходимости выбора одного из нескольких устройств. Select Source (выбор источника) вызывает Data Manager, поддерживающий диалог выбора устройства ввода. Когда выбор сделан, Source Manager обеспечивает процесс "capabilities negotiation" (возможности ведения переговоров), который происходит между прикладной программой, посылающей сообщение с описанием ожидаемых данных, и источником ввода данных (источником дается определение данных, которые он может передавать программе). Когда "переговоры" закончены, все управление интерфейсом пользователя передается источнику ввода данных. Далее пользователю предоставляется возможность установить параметры устройства и начать сканирование. Источник ввода данных передает изображения обратно источнику управления данными, откуда они попадут в прикладную программу, которая позволит пользователю разместить изображение в публикации.

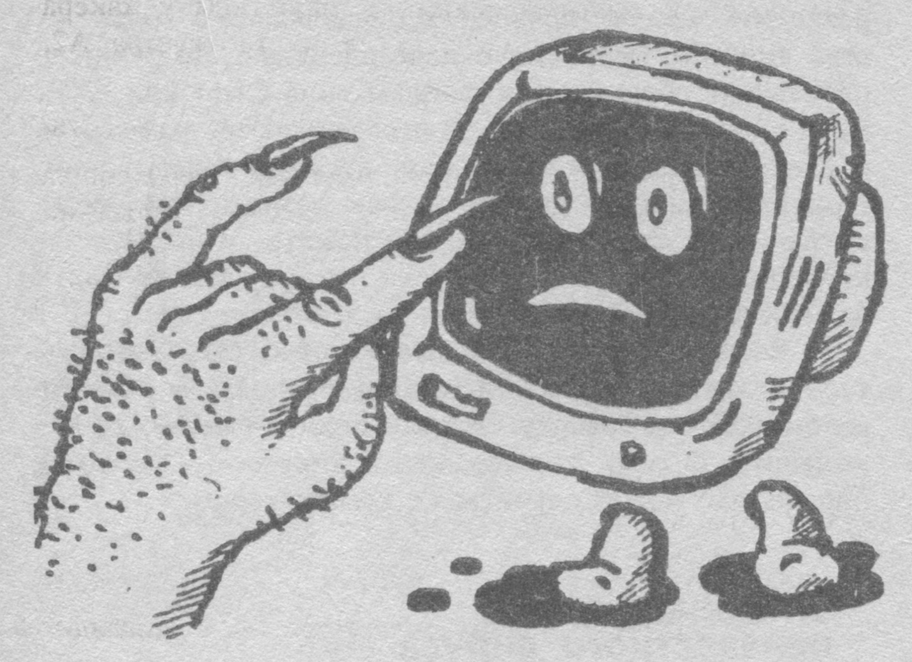
Как не оказаться в сложной ситуации

Пользователю следует обращать особое внимание на наличие слова TWAIN на покупаемом аппаратном и программном обеспечении и интересоваться у поставщиков продуктов, когда они намерены поддерживать TWAIN.

Первым сканером, поддерживающим TWAIN, стал HP ScanJet IIp. Благодаря применению новых возможностей этого интерфейса он позволяет пользователю сканировать изображение одновременно с работой в прикладной программе.

Otto Bauer


EGA
640×350
VGA
640×480
SuperVGA
1024×768



Как и следовало ожидать, на территории бывшего СССР разразилась настоящая эпидемия компьютерной заразы, принимающей самые разнообразные и довольно изощренные формы. Телефонные звонки от обеспокоенных владельцев персональных компьютеров раздаются ежедневно из самых разных мест — от Украины до Благовещенска и обсерваторий в Казахстане. В отличие от традиционного гриппа, эти эпидемии все реже заносятся "оттуда" и все чаще имеют ярко выраженные отечественные приметы ...

Прежде чем лечить доктора

— Доктор, я умру ?
— Обязательно !!!

Наши создатели вирусов недолго плелись в хвосте мирового компьютерного хулиганства — пропорционально числу ввозимых компьютеров росло и поголовье электронных насекомых, неизвестных знаменитым SCAN'ам и TNTVIRUS'ам. И, как следствие, практически все пользователи компьютеров рано или поздно (а кое-кто и постоянно) встречаются с вирусом. И, как еще одно следствие, далеко не всегда антивирусные программы правильно опознают этот вирус, и, что гораздо важнее, далеко не все антивирусы правильно его удаляют.

Естественно, что если за соседним столом работает системный программист, который за 2-3 часа выловит гостя и стерилизует компьютер (не прибегая при этом к команде "FORMAT C:"), или вы сами являетесь таковым, то можно спать спокойно. Но если это не так, то придется либо самому становиться системным программистом, либо выделить вирус и отослать его разработчику (или разработчикам) антивирусных программ, а затем с нетерпением ждать ответа.

Ниже приведены рекомендации пользователям, оказавшимся в подобной ситуации. Естественно, что эти рекомендации отражают только мой личный опыт, и специалисты в области вирусов, особенно их авторы, могут со мной поспорить. В первую очередь, описаны правила использования антивирусных программ, затем методы выделения вируса и наиболее общие способы анализа и удаления вируса. Часть материала, приведенного ниже, впервые опубликована в журнале "Интеркомпьютер", № 6, 1990 г.

1. Методика использования антивирусных программ

Перед использованием антивирусных программ крайне желательно загрузиться с резервной копии DOS, которая расположена на заведомо чистой от вирусов и защищенной от записи дискете. Это необходимо для того, чтобы застраховаться от присутствия резидентного вируса, так как он может блокировать работу антивируса или использовать его работу для инфицирования проверяемых файлов. Перезагрузка компьютера должна быть холодной, так как некоторые вирусы "выживают" при теплой перезагрузке.

Желательно, чтобы антивирусные программы, используемые для проверки, были самых последних версий. Желательно также, чтобы хоть одна из них была отечественного производства, так как процесс эмиграции вируса в другие страны и иммиграции антивирусных программ занимает довольно большое время. Поэтому наиболее оперативно на появление нового вируса реагируют только отечественные антивирусные программы.

Если обнаружены зараженные файлы, то следует:
- распечатать их список;
- если для этих файлов нет backup-копии, то сохранить их на дискеты;
- при помощи антивирусной программы восстановить зараженные файлы и затем проверить их работоспособность и соответствие backup-копии (если есть);
- если восстановление файлов произошло не вполне корректно, то их следует уничтожить и переписать с backup-копий; если же копий нет, то восстановить зараженные файлы с дискет и попытаться дезактивировать их при помощи другого антивируса.

Следует отметить, что качество восстановления файлов многими антивирусными программами оставляет желать лучшего. Многие популярные антивирусы частенько необратимо портят файлы вместо их лечения. Поэтому если потеря файлов нежелательна, то выполнять перечисленные выше пункты следует в полном объеме.

Необходимо обращать особое внимание на чистоту модулей, сжатых утилитами типа LZEXE, PKLITE или DIET, файлов в архивах (ZIP, ARC, ICE, ARJ и т.д.) и данных в самораспаковывающихся файлах, созданных утилитами типа ZIP2EXE. Если случайно упаковать файл, зараженный вирусом, то обнаружение и удаление такого вируса без распаковки файла практически невозможно. В данном случае типичной будет ситуация, при которой все антивирусные программы сообщат о том, что от вирусов очищены все диски, но через некоторое время вирус появится опять.

Штаммы вируса могут проникнуть и в backup-копии программного обеспечения при обновлении этих копий. Причем архивы и backup-копии являются основными поставщиками давно известных вирусов. Вирус может годами "сидеть" в дистрибутивной копии какого-либо программного продукта и неожиданно проявиться при установке программ на новом компьютере.

Никто не может гарантировать полного уничтожения всех копий компьютерного вируса, так как файловый вирус может поразить не только выполняемые файлы, но и оверлейные модули с расширениями имени, отличающимися от COM или EXE. Загрузочный вирус может остаться на какой-либо дискете и внезапно проявиться при случайной попытке перезагрузить с нее. Поэтому целесообразно некоторое время после удаления вируса постоянно пользоваться резидентным антивирусным монитором типа утилиты -D.COM, входящей в комплекс "«Доктор» Касперский". При помощи резидентного монитора можно перехватить момент появления подавляющего большинства вирусов. Если вирус нерезидентный, то монитор перехватит обращение к файлам типа COM или EXE. Если вирус резидентный, то отслеживаются изменения в блоках памяти и в таблице векторов прерываний. При этом следует выяснить, как резидентный вирус выглядит на карте оперативной памяти, и после дезактивации вируса периодически просматривать карту памяти на предмет обнаружения этого вируса.

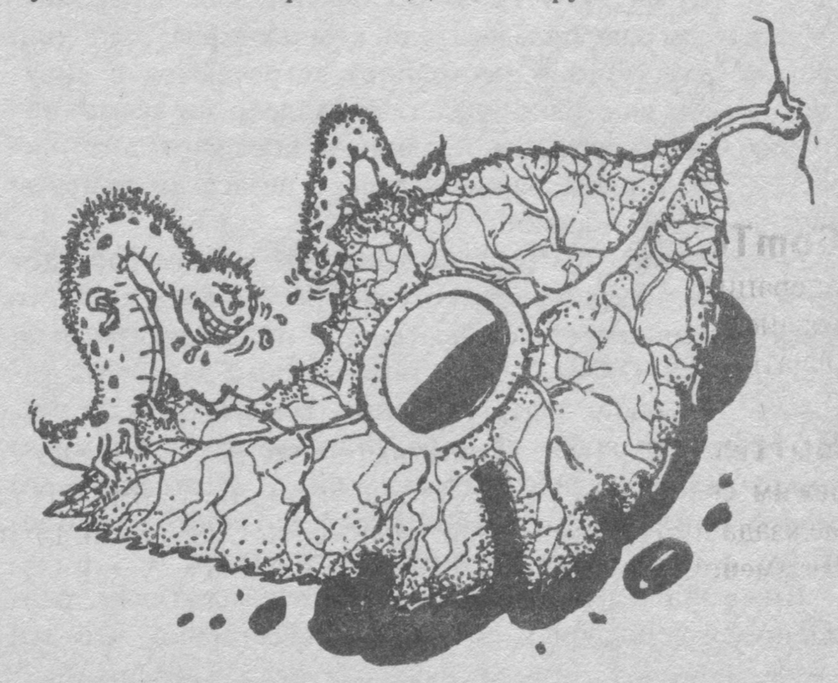
И последнее замечание. Не каждый сбой компьютера является проявлением вируса, поэтому почаще вспоминайте поговорку "Не так страшен черт, как его малюют". К тому же, поражение вирусом — не самое плохое, что может случиться с компьютером.

2. Обнаружение неизвестного вируса

В этом разделе рассматриваются ситуации, с которыми может столкнуться пользователь в том случае, если его компьютер поражен вирусом, но ни одна из известных ему антивирусных программ не дала положительного результата. Где и как искать вирус? Какие при этом необходимы инструментальные средства и какими методами следует пользоваться?

2.1. Обнаружение файлового вируса

Как известно, вирусы делятся на резидентные и нерезидентные. Встречавшиеся мне до сих пор резидентные вирусы отличались гораздо большим коварством и изощренностью, чем нерезидентные. Поэтому для начала рассмотрим простейший случай — поражение компьютера неизвестным нерезидентным вирусом. Такой вирус активизируется при запуске какой-либо зараженной программы, совершает все, что ему положено, передает управление программе-носителю и в дальнейшем (в отличие от резидентных вирусов) не будет мешать ее работе. Для обнаружения такого ви-



руса необходимо сравнить длины файлов на винчестере и в дистрибутивных копиях (упоминание о важности хранения таких копий уже стало банальностью). Если это не поможет, то побайтно сравнить дистрибутивные копии с используемыми программами. В настоящее время разработано достаточно много утилит такого сравнения файлов, самая простейшая из них (утилита COMP) содержится в DOS.

Существуют специальные утилиты, которые не только сравнивают содержимое файлов и их параметров (длина, атрибуты, время последней модификации), но и сообщают о наличии в разных файлах повторяющихся одинаковых участков кода или проверяют файлы на наличие "подозрительных" команд (например, утилиты Virus Locator А.Шеховцова или Lie Detector Е.Сусликова, алгоритм теста на "подозрительность" включен и в программу-фаг -V.EXE). Такие утилиты очень удобны, но иногда начинают "ругаться" на широко распространенные упаковщики EXE-файлов, некоторые файлы DOS, на антивирусные "насадки" к файлам, плохо обнаруживают вирусы, часть кода которых зашифрована. К тому же, в последнее время появляется все большее количество вирусов-"призраков", у которых два штамма одного вируса практически никогда не имеют ни одного совпадающего байта (в большинстве случаев это достигается сложными шифровками тела вируса и многочисленными модификациями кодов в блоке расшифровки).

Можно также просмотреть дамп выполняемых файлов. В некоторых случаях можно сразу обнаружить присутствие вируса по наличию в его коде текстовых строк. Многие вирусы, например, содержат строки: ".COM", "*.COM", "*.*", "MZ", "COMMAND" и т.д. Эти строки часто встречаются в начале или в конце зараженных файлов. Существует и еще один способ визуального определения зараженного вирусом файла. Он основан на том, что выполняемые файлы, исходный текст которых написан на языке высокого уровня, имеют вполне определенную структуру: сегмент кода программы, а сразу за ним — сегмент данных, причем в начале этого сегмента стоит строка-копирайт фирмы-изготовителя компилятора. Если в дампе такого файла за сегментом данных следует еще один участок кода, то вполне вероятно, что файл заражен вирусом.

Рекомендуется запустить одну из резидентных антивирусных программ-мониторов и следить за ее сообщениями о "подозрительных" действиях программ (запись в COM- или EXE-файлы, запись на диск по абсолютному адресу и т.п.). Существуют мониторы, которые не только перехватывают такие действия, но и сообщают адрес, откуда поступил "подозрительный" вызов (к таким мониторам относится -D.COM). Обнаружив подобное сообщение, следует выяснить, от какой программы оно пришло, и проанализировать ее коды при помощи резидентного дизассемблера (например, -UTIL.COM). При анализе кодов программ, резидентно находящихся в памяти, большую помощь часто оказывает трассирование прерываний 13h и 21h.

2.2. Обнаружение загрузочного вируса

В загрузочных секторах дисков расположены небольшие, как правило, программы, назначение которых состоит в определении размеров и границ логических дисков (для MBR-винчестера) или загрузке операционной системы (для Boot-сектора). Дампы таких программ приведены в Приложении.

Вначале следует прочитать содержимое сектора, "подозрительного" на наличие вируса. Для этой цели лучше использовать программы из комплекта "Нортоновских утилит".

Некоторые загрузочные вирусы практически сразу обнаруживают свое присутствие различными текстовыми строками (например, вирус "Stone" содержит

строки: "Your PC is now Stoned!", "LEGALISE MARIJUANA!"). Некоторые вирусы, поражающие Boot-секторы дисков, проявляются отсутствием строк, обязательно присутствующих в Boot-секторе. К таким строкам относятся: строка-заголовок Boot-сектора, содержащая номер версии DOS или название фирмы-производителя программного обеспечения (например, строка "MSDOS3.3"); имена системных файлов (например, строка "IO SYSMSDOS SYS"); строки сообщений об ошибках.

Стандартный загрузчик MS-DOS, расположенный в MBR, занимает меньше половины сектора, и многие вирусы, поражающие MBR винчестера, довольно просто заметить по увеличению длины кода, расположенного в секторе MBR.

Однако существуют вирусы, которые внедряются в загрузчик без изменения его текстовых строк и с минимальными изменениями кода загрузчика (например, вирусы, изменяющие при инфицировании MBR всего 3 байта Disk Partition Table, соответствующие адресу активного загрузочного сектора). Для идентификации такого вируса придется провести более детальное исследование кодов загрузочного сектора вплоть до полного анализа алгоритма работы его кода.

Приведенные рассуждения основываются на том, что стандартные загрузчики (программы, записываемые операционной системой в загрузочные секторы) реализуют стандартные алгоритмы загрузки операционной системы и оформляются в соответствии с ее стандартами. Если же диски отформатированы утилитами, не входящими в состав DOS (например, Disk Manager), то для обнаружения в них вируса следует проанализировать алгоритм работы и оформление загрузчиков, создаваемых такой утилитой.

Рассмотренные выше методы обнаружения файловых и загрузочных вирусов подходят для большинства как резидентных, так и нерезидентных вирусов. Однако эти методы не срабатывают, если вирус выполнен по технологии "стелс", что делает бесполезным использование большинства резидентных мониторов, утилит сравнения файлов и чтения секторов.

2.3. Обнаружение резидентной части вируса

Если в компьютере обнаружены следы деятельности вируса, но видимых изменений в файлах и системных секторах дисков не наблюдается, то вполне возможно, что компьютер поражен одним из "стелс"-вирусов. В этом случае необходимо загрузить DOS с заведомо чистой от вирусов дискеты, содержащей резервную копию DOS, и действовать как и при поражении нерезидентным вирусом. Однако иногда это нежелательно, а в ряде случаев невозможно (известны, например, случаи покупки новых компьютеров, зараженных вирусом). Тогда придется обнаружить и нейтрализовать резидентную часть вируса, выполненную по технологии "стелс". Возникает вопрос: где в памяти и как искать вирус или его резидентную часть? В настоящее время известно несколько способов инфицирования памяти.

1. Вирус может проникнуть в таблицу векторов прерываний. Лучший способ обнаружить такой вирус состоит в том, чтобы просмотреть карту распределения памяти, которая отображает список резидентных программ (пример такой карты приведен в табл. 1). Подробная карта памяти сообщает информацию о всех блоках, на которые разбита память: адрес блока управления памятью MCB, имя программы — владельца блока, адрес ее префикса программного сегмента (PSP) и список перехватываемых блоком векторов прерываний.

При наличии вируса в таблице векторов прерываний утилиты, отображающие карту распределения памяти (например, -D.COM, -UTIL.COM), начинают "шуметь".

Таблица 1. Карта распределения незараженной памяти

Адрес блока MCB	Адрес PSP	Размер блока MCB(Кб)	Имя программы владельца блока MCB	Номера векторов прерываний
0E9A	0E9B	3,20K	COMMAND.COM	22,24,2E
0F6E	0F6F	0,04K	блок свободен	
0F72	0E9B	0,15K	COMMAND.COM	
0F7D	0F85	23,20K	-D.COM	10,1B,1C,23,26,27 28,2F,40,EF,FF
1545	1546	0,10K	блок свободен	
154D	154E	3,90K	PLUCK.COM	09,13,16,21
164B	0E9B	549,30K	COMMAND.COM	30,EE,F2,F3,F4,F5 F6,F7,F8,FE

Таблица 2. Таблица векторов прерываний поражена вирусом

Адрес блока MCB	Адрес PSP	Размер блока MCB(Кб)	Имя программы владельца блока MCB	Номера векторов прерываний	
0E9A	0E9B	3,20K	COMMAND.COM	22,24,2E,EB	
0F6E	0F6F	0,04K	блок свободен		
0F72	0E9B	0,15K	COMMAND.COM		
0F7D	0F86	23,20K	-D.COM	10,1B,23,26,27,28 2F,40,CA,D3	
1546	1547	0,10K	блок свободен		
154E	154F	3,90K	PLUCK.COM	09,13,16,91	
164C	0E9B	549,30K	COMMAND.COM	30,85,89,8E,90,93 95,97,98,9D,9E,9F A2,A7,A9,AB,AE,AF B1,B3,BB,BE,BF,C0 C4,C9,CE,CF,D0,D1 D2,D4,D5,D8,D9,DA DB,DD,DF,E0,E4,E5 E6,E7,E8,E9,EA,ED EE,F3,F4,F5,F7,F8 F9	"Шум"

Другой, более надежный, но требующий высокой квалификации пользователя способ, — просмотреть таблицу векторов прерываний с помощью дизассемблера. Если при этом будут обнаружены коды какой-то программы, то код вируса (или участок кода) найден.

2. Вирус может несколькими способами встроиться в DOS: в произвольный системный драйвер, в системный буфер, в другие рабочие области

DOS (например, в область системного стека или в свободные места таблиц DOS и BIOS). Пока известен единственный способ инфицирования вирусом произвольного системного драйвера: прикрепление тела вируса к файлу, содержащему драйвер, и модификация заголовка поражаемого драйвера. Если при этом вирус оформляет себя как отдельный драйвер, то его можно обнаружить при просмотре карты распределения памяти, содержащей список



системных драйверов. Если при этом в списке присутствует драйвер, который не описан в файле CONFIG.SYS, то он и может быть вирусом. Если же вирус "приклеивается" к расположенному перед ним драйверу, не выделяя свои коды как отдельную программу-драйвер, то обнаружить его можно методами, описанными ниже.

Вирус, встраивающийся в системный буфер, должен уменьшать общее число буферов; в противном случае он будет уничтожен последующими операциями считывания с диска. Достаточно несложно написать программу, которая подсчитывает число буферов, реально присутствующих в системе, и сравнивает полученный результат со значением команды BUFFERS, расположенной в файле CONFIG.SYS (если команда BUFFERS отсутствует, то со значением, устанавливаемым DOS по умолчанию). Подобная проверка прекрасно работает (100% попаданий) в антивирусной программе -V.EXE.

Существует достаточно способов внедрения вируса в системные таблицы или область стека DOS. Однако реализация этих способов потребует от автора вируса досконального знания различных версий DOS. К тому же свободного места в DOS не так уж много, поэтому написание полноценного "стелс"-вируса такого типа маловероятно. Если же все-таки подобный вирус появится, то обнаружить его код можно дизассемблированием "подозрительных" на наличие вируса участков DOS.

3. Вирус может проникнуть в область программ в виде:

- отдельной резидентной программы или отдельного блока памяти (MCB);
- внутри или "приклеившись" к какой-либо резидентной программе.

Если вирус внедряется в отведенную для прикладных программ область памяти в виде нового блока, создавая для себя собственный MCB, или как отдельная резидентная программа, то его можно обнаружить при просмотре подробной карты распределения памяти, отображающей адреса всех блоков MCB. Обычно такой вирус выглядит как отдельный блок памяти (табл. 3), не имеющий имени и перехватывающий один или несколько векторов прерываний (например, int 8h, 13h, 1Ch, 21h).

Таблица 3. Вирус в области пользовательских программ

Адрес блока MCB	Адрес PSP	Размер блока MCB(Кб)	Имя программы владельца блока MCB	Номера векторов прерываний
0E9A	0E9B	3,20K	COMMAND.COM	22,24,2E
0F6E	0F6F	0,04K	блок свободен	
0F72	0E9B	0,15K	COMMAND.COM	
10D1	10DA	23,20K	-D.COM	10,1B,23,26,27,28 2F,40
17AF	17B0	1,70K	?	08
1820	0E9B	539,30K	COMMAND.COM	30,31
9F04	0000	3,90K	?	1C,21

Обнаружен вирус "Yankee Doodle".
Обнаружен вирус "Jerusalem".

4. Вирус может проникнуть за границу памяти, выделенной для DOS. Все загрузочные и некоторые файловые вирусы располагаются за пределами памяти, выделенной для DOS, уменьшая значение слова, расположенного по адресу [0040:0013]. Обнаружить такие вирусы очень просто — достаточно узнать емкость оперативной памяти и сравнить ее с реальной. Если вместо 640 или 512 Кбайт система сообщит меньшее значение, то следует просмотреть дизассемблером "отрезанный" участок памяти. Если на этом участке будут обнаружены коды какой-то программы, то, скорее всего, вирус найден.

Внимание!

Емкость оперативной памяти может уменьшиться на один или несколько килобайт и в результате использования расширенной памяти или некоторых типов контроллеров. При этом типична следующая картина: в "отрезанном" участке содержимое большинства байтов нулевое.

5. Вирус может встраиваться в конкретные, заведомо резидентные программы или "приклеиться" к уже имеющимся блокам памяти. Возможно инфицирование вирусом файлов DOS, которые являются резидентными (например, IO.SYS, MSDOS.SYS, COMMAND.COM), загружаемых драйверов (ANSY.SYS, COUNTRY.SYS, RAMDRIVE.SYS) и др. Обнаружить такой вирус гораздо сложнее вследствие малой скорости его распространения, но, однако, вероятность атаки подобного вируса значительно меньше. Все чаще стали встречаться "хитрые" вирусы, которые корректируют заголовки блоков памяти или "обманывают" DOS таким образом, что блок с кодами

вируса становится одним целым с предыдущим блоком памяти.

В этом случае обнаружить вирус гораздо сложнее — необходимо знать реальную длину программ, размещенных в памяти, и список прерываний, которые они перехватывают. Но этот способ не очень удобен и иногда не срабатывает. Поэтому рекомендуется использовать другой метод, который может облегчить выявление вируса в подобной ситуации. Он основан на следующем свойстве — подавляющее большинство вирусов для обнаружения незараженных файлов или секторов дисков перехватывают прерывания 13h или 21h, встраиваясь в обработчик прерывания. В таком случае для обнаружения вируса достаточно просмотреть текст (команды ассемблера) обработчиков указанных прерываний (например, при помощи программы -UTIL.COM). Правда, для того, чтобы отличить вирус от обычных программ, требуется достаточный опыт работы с вирусами и некоторое представление о структуре обработчика на незараженном компьютере. К тому же следует быть осторожным: существует несколько вирусов, которые "завешивают" систему при попытке трассировки их кодов.

Известны вирусы, которые при заражении файлов или дисков не пользуются прерываниями, а напрямую работают с ресурсами DOS. При поиске подобного вируса необходимо тщательно исследовать изменения во внутренней структуре зараженной DOS: список драйверов, таблицы файлов, стеки DOS и т.д. Это очень кропотливая работа, которая, учитывая многочисленность версий DOS, требует очень высокой квалификации пользователя.

Конечно, возможны и другие, достаточно экзотические способы инфицирования памяти вирусом, например, внедрение вируса в видеопамять (что уже встречалось) или в расширенную память, но подобные вирусы встречаются достаточно редко и всегда проявлялись хотя бы одним из перечисленных выше признаков. Возможно, конечно, создание монстров, использующих виртуальные режимы процессоров 80386 и 80486 (а что будет в 80586?), но, во-первых, это будет слишком заметно, а во-вторых, программист подобной квалификации всегда найдет для себя более осмысленную задачу.

3. Анализ алгоритма вируса

На мой взгляд, наиболее удобным для хранения и анализа вируса объектом является файл, содержащий его (вируса) тело. Как показывает практика, для анализа файлового вируса удобнее иметь несколько зараженных файлов различных, но не очень больших, длин. При этом желательно иметь зараженные файлы всех типов (COM, EXE и SYS), которые могут быть поражены вирусом. Если необходимо проанализировать часть оперативной памяти, то при помощи некоторых утилит (например, -UTIL.COM) довольно просто выделить участок, где расположен вирус, и скопировать этот участок на диск. Если же требуется анализ сектора MBR или Boot-сектора, то скопировать их в файлы можно при помощи популярных "Нортоновских утилит". Для хранения загрузочного вируса наиболее удобен файл-образ зараженного диска. Для его получения необходимо отформатировать дискету (лучше 360 Кбайт), заразить ее вирусом, скопировать образ дискеты (все сектора, начиная с 0-го и кончая последним) в файл и, при необходимости, скомпрессировать его (эту процедуру можно проделать при помощи "Нортоновских утилит", программ TELEDISK или DISKDUPE).

Зараженные файлы или файл-образ зараженной дискеты лучше передать разработчикам антивирусных программ по модему или, в крайнем случае, на дискете по почте в обмен на антивирусную программу. Однако, если это займет много времени, которое, как известно, не ждет, то пользователям, достаточно уверенным в себе, можно попробовать и самостоятельно разобраться в вирусе и написать антивирус.

При анализе алгоритма вируса предстоит выяснить:
- способ(ы) размножения вируса;
- характер возможных повреждений, которые вирус нанес информации, хранящейся на дисках;
- метод лечения оперативной памяти и зараженных файлов (секторов).

При решении этих задач не обойтись без дизассемблера или отладчика (это могут быть отладчики Quaid Analyzer, Advanced Trace86, FULSCREEN DEBUG или дизассемблеры DisDoc, Sourcer или IDA).

И отладчики, и дизассемблеры имеют и положительные, и отрицательные черты — каждый выбирает то, что он считает более удобным. Несложные короткие вирусы быстро "вскрываются" стандартным отладчиком DEBUG; при анализе объемных и высокосложных (типа вируса "V-4096") не обойтись без дизассемблера. Если необходимо быстро обнаружить метод восстановления пораженных файлов, то достаточно пройтись отладчиком по началу вируса до того места, где он восстанавливает загруженную программу перед тем, как передать ей управление (фактически именно этот алгоритм чаще всего используется при лечении вируса). Если же требуется получить детальную картину работы вируса или хорошо документированный листинг, то, кроме дизассемблера Sourcer и его возможности восстанавливать перекрестные ссылки, здесь вряд ли что поможет. К тому же, следует учитывать, что, во-первых, некоторые вирусы достаточно успешно блокируют попытки протрассировать их коды, а во-вторых, при работе с отладчиком существует ненулевая вероятность того, что вирус вырвется из-под контроля.

При анализе файлового вируса необходимо выяснить, какие файлы (COM, EXE, SYS) поражаются вирусом, в какое место (места) в файле записывается код вируса — в начало, конец или середину файла, в каком объеме возможно восстановление файла (полностью или частично), в каком месте вирус хранит восстанавливаемую информацию.

При анализе загрузочного вируса основная задача — выяснение адреса (адресов) сектора, в котором вирус сохраняет первоначальный загрузочный сектор (если, конечно, вирус сохраняет его).

Для резидентного вируса также необходимо выделить участок кода, создающий резидентную копию вируса, и вычислить возможные адреса точек входа в перехватываемые вирусом прерывания. Необходимо также выяснить, каким образом и где в оперативной памяти выделяет вирус место для своей резидентной копии: записывается ли вирус по фиксированным адресам в системные области DOS и BIOS, уменьшает размер памяти, выделенной под DOS (слово по адресу [0000:0413]), создает для себя специальный MCB-блок, либо использует какой-либо другой способ.

Существуют особые случаи, когда анализ вируса может оказаться очень сложной для пользователя задачей, например, при анализе вируса-"призрака". В этом случае лучше обратиться к специалисту по анализу кодов программ.

4. Восстановление пораженных объектов

При поражении компьютера произвольным вирусом перед пользователем могут возникнуть следующие задачи: дезактивация резидентной памяти, лечение загрузочных секторов (MBR, Boot), обход подкаталогов всех дисков, поиск и восстановление зараженных файлов.

4.1. Дезактивация оперативной памяти

При лечении оперативной памяти следует обнаружить коды вируса и изменить их таким образом, чтобы вирус в дальнейшем не мешал работе антивирусной программы.

Оптимальный алгоритм обнаружения кодов вируса зависит от способа инфицирования вирусом оперативной памяти. Если вирус записывает свою резидентную копию по фиксированным адресам, то и искать ее нужно по этим адресам. Если вирус создает для своей TSR-копии новый MCB-блок, то следует пройти все MCB-блоки, начиная с первого и кончая последним, и в каждом блоке по фиксированным адресам искать коды вируса. Если вирус записывается за пределами памяти DOS, предварительно уменьшая размер выделенной под DOS памяти, то следует провести полное сканирование "отрезанного" от DOS участка. Практически для всех способов инфицирования вирусом оперативной памяти подходит метод трассировки прерываний, используемый вирусами семейства "Yankee".

После того как определены адреса TSR-копии вируса, следует исправить его коды таким образом, чтобы полностью нейтрализовать влияние вируса на процесс поиска и лечения зараженных файлов. Итак, если, например, вирус заражает файлы при их открытии, это может выглядеть примерно следующим образом:

```
Исходный код вируса                 Код дезактивированного вируса
. . . .     . . . . .               . . . .     . . . . .
80 FC 3D    CMP   AH,3Dh            80 FC 3D    CMP   AH,3Dh
74 xx       JE    Infect_File       90          NOP
E9 xx xx    JMP   Continue          90          NOP
. . . .     . . . . .               E9 xx xx    JMP   Continue
                                    . . . .     . . . . .
```

При дезактивации TSR-копии вируса следует учитывать, что вирус может предпринимать специальные меры для восстановления своих кодов (например, некоторые вирусы семейства "Yankee"), и в этом случае придется нейтрализовать и механизм самовосстановления вируса.

4.2. Восстановление загрузочных секторов

Восстановление секторов в большинстве случаев довольно простая операция, выполняемая при помощи "Нортоновских утилит" или DOS´овской команды SYS. При заражении компьютера вирусом со сложным алгоритмом основной задачей становится определение адреса сектора, в котором находится сохраненный вирусом оригинальный загрузочный сектор. Некоторые вирусы не сохраняют старое содержимое сектора (вирус "Azusa"); в этом случае придется восстанавливать коды загрузочного сектора по какому-либо образцу (например, по образцам, приведенным в Приложении).

При восстановлении системных загрузчиков следует соблюдать особую осторожность, поскольку результатом неправильного исправления сектора MBR или Boot-сектора может быть потеря всей информации на диске (дисках). Поэтому даже довольно простую процедуру считывания и восстановления оригинального загрузочного сектора нужно обязательно проверить на корректность. Простейшая проверка состоит в выяснении наличия в конце загрузочного сектора сигнатуры AA55h.

4.3. Восстановление файлов

При лечении файлов следует учесть следующие правила:
- необходимо протестировать и вылечить все выполняемые файлы (COM, EXE, SYS, оверлеи) во всех каталогах всех дисков вне зависимости от атрибутов файлов (т.е. файлы read-only, системные и скрытые), а также выполняемые файлы, упакованные в архивах;
- желательно сохранить неизменными атрибуты и дату последней модификации файла;
- следует учесть возможность многократного поражения файла вирусом ("бутерброд" из вирусов).

Само лечение файла производится в большинстве случаев довольно просто. Если вирус записывается в

конец файла, то необходимо восстановить старое начало файла (у EXE-файлов особое внимание следует обратить на корректность полей восстанавливаемых заголовков) и затем отрезать вирус от файла (пример программы, восстанавливающей файлы, пораженные вирусом “Vienna-648“, приведен в Приложении). Если вирус записался в начало файла, то следует переписать старое содержимое файла в его (файла) начало и укоротить файл.

В более сложных случаях (особенно, если вирус зашифрован или выполнен по технологии “призрак“) процедура восстановления файла может существенно усложниться.

Е.Касперский

Москва, Тессинский пер. 9/16, АО “Ками“
Телефоны (095) 499-15-00, 928-21-38
E-mail: eugene@kami.npimsu.msk.su

Термины

Дизассемблер — утилита, осуществляющая преобразование, обратное ассемблированию, то есть переводящая машинные коды в язык ассемблера. Такие утилиты крайне необходимы не только при отладке программ (для чего они и создаются), но и при анализе вируса.

Дизассемблирование — перевод машинных кодов какой-либо программы в ее представление на языке ассемблера.

Монитор (программа-монитор, блокировщик) — резидентно находящаяся в оперативной памяти утилита, которая позволяет выявлять “подозрительные“ действия пользовательских программ: изменение и переименование выполняемых программ (COM- и EXE-файлов), запись на диск по абсолютному адресу, форматирование диска и т.д. При обнаружении “подозрительной“ функции программа-монитор либо выдает на экран сообщение, либо блокирует выполнение перехваченной функции, либо совершает другие специальные действия.

Призрак (вирусы-“призраки“) — вирусы, предпринимающие специальные меры для затруднения их поиска и анализа. Не имеют сигнатур, то есть не содержат ни одного постоянного участка кода. В большинстве случаев два образца одного и того же вируса-“призрака“ не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика.

Стелс (Stealth) — “стелс“-вирусы (вирусы-невидимки) представляют собой программы, которые перехватывают обращения DOS к пораженным файлам или секторам дисков и “подставляют“ вместо себя незараженные участки информации. Кроме этого, такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие “обманывать“ резидентные антивирусные мониторы. К “стелс“-вирусам относятся вирусы “V-4096“, “Fish #6“, “Brain“ и некоторые другие.

Программа-доктор для файла, пораженного вирусом "Vienna-648"

```
        Offset_3_Bytes=206h

SEGA    SEGMENT BYTE PUBLIC
        ASSUME  CS:SEGA, DS:SEGA

        ORG     100h

CURE    PROC
START:
        PUSH    CS                      ;Код и данные
        POP     DS                      ; в одном сегменте

        MOV     DX, OFFSET File_Name    ;Открыть файл
        MOV     AX,3D02h
        INT     21h
        JC      Error

        MOV     Handle,AX

        MOV     CX,3                    ;Прочитать первые
        MOV     DX, OFFSET Buffer       ; 3 байта файла
        MOV     BX,Handle
        MOV     AH,3Fh
        INT     21h
        JC      Error

        XOR     CX,CX                   ;Вычислить адрес
        MOV     AX, WORD PTR Buffer+1   ; первоначального
        ADD     AX, Offset_3_Bytes      ; начала файла
        ADC     CX,0                    ; (старые 3 байта)

        MOV     DX,AX                   ;Установить указатель
        MOV     BX,Handle               ; на старые 3 байта
        MOV     AX,4200h
        INT     21h
        JC      Error

        MOV     CX,3                    ;Прочитать старое
        MOV     DX, OFFSET Buffer+3     ; начало файла
        MOV     BX,Handle
        MOV     AH,3Fh
        INT     21h
        JC      Error

        XOR     CX,CX                   ;Переместить указатель
        XOR     DX,DX                   ; на начало файла
        MOV     BX,Handle
        MOV     AX,4200h
        INT     21h
        JC      Error

        MOV     CX,3                    ;Восстановить начало файла
        MOV     DX, OFFSET Buffer+3
        MOV     BX,Handle
        MOV     AH,40h
        INT     21h
        JC      Error

        XOR     CX,CX                   ;Переместить указатель
        MOV     DX, WORD PTR Buffer+1   ; на начало вируса
        ADD     DX,3
        ADC     CX,0
        MOV     BX,Handle
        MOV     AX,4200h
        INT     21h
        JC      Error

        MOV     CX,0                    ;Укоротить файл
        MOV     BX,Handle
        MOV     AH,40h
        INT     21h

Error:
        MOV     BX,Handle               ;Закрыть файл
        MOV     AH,3Eh

        MOV     AX,4C00h                ;Выход в DOS
        INT     21h

Handle     DW   ?
File_Name  DB   "INFECT.COM"
Buffer     DB   6 DUP (?)

CURE       ENDP
SEGA       ENDS
           END  START
```

Приложение

Дамп сектора, содержащего MBR

```
С адреса 0000 по адрес 00E1 -- программа загрузки активного Boot-сектора.
С адреса 01C0 по адрес 01FD -- таблица разбиения диска (Disk Partition Table).

0000  FA2BC08E D08EC08E D8B8007C 8BE0FB8B  F0BF007E FCB90001 F3A5E900 02B91000  "·+└O╨O└O>┐.|Лр√Л≡┐.~"▌..≤ещ..╡.."
0020  8B36857E F6048075 0883EE10 E2F6EB37  90BFBE07 57B90800 F3A55EBB 007C8B14  "Л6Е~÷.Au.Гю.т÷ы7Р┐┘.W▌..≤e^┐.|Л."
0040  8B4C02BD 0500B801 02CD1373 092BC0CD  134D7419 EBF0BEFE 7DAD3D55 AA7514BE  "ЛL.╜..┐..═.s.+└═.Mt.ы≡┘■}н=Uku.┘"
0060  BE07EA00 7C00008B 36877EEB 0A8B3689  7EEB048B 368B7EAC 0AC074FE BB0700B4  "┘.ъ.|..Л63~ы.Л6Й~ы.Л6Л~м.└t■┐..┤"
0080  0ECD10EB F2EE7F8D 7EA77EC8 7E0D0A49  6E76616C 69642050 61727469 74696F6E  ".═.ы≥ю.Н~з~╚~..Invalid Partition"
00A0  20546162 6C65000D 0A457272 6F72204C  6F616469 6E67204F 70657261 74696E67  " Table...Error Loading Operating"
00C0  20537973 74656D00 0D0A4D69 7373696E  67204F70 65726174 696E6720 53797374  " System...Missing Operating Syst"
00E0  656D0000 00000000 00000000 00000000  00000000 00000000 00000000 AA550000  "em..........................кU.."
0100  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0120  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0140  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0160  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0180  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01C0  01010405 22C8CC00 0000609F 00000000  01C90505 E22B2CA0 0000E4E6 01000000  "...."╚╠...`Я.....╔..т+,а..фц...."
01E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 000055AA  "..............................Uк"
```

Дамп сектора, содержащего нормальный Boot-сектор

```
0000  EB34904D 53444F53 332E3300 02040100  02000260 9FF82800 22000600 CC000000  "ы4PMSDOS3.3........`Я°(."...╠..."
0020  00000000 00000000 00000000 00000012  00000000 0100FA33 C08ED0BC 007C1607  "........................3└O╨╝.|.."
0040  BB780036 C5371E56 1653BF2B 7CB90B00  FCAC2680 3D007403 268A05AA 8AC4E2F1  "┐x.6┼7.V.S┐+|╣.."м&А=.t.&К.кК-т±"
0060  061F8947 02C7072B 7CFBCD13 7267A010  7C98F726 167C0306 1C7C0306 0E7CA33F  "..ЙG.╟.+|√═.rgа.|Ш≈&.|...|...|г?"
0080  7CA3377C B82000F7 26117C8B 1E0B7C03  C348F7F3 0106377C BB0005A1 3F7CE89F  "|г7|┐ .≈&.|Л..|.├H≈≤..7|┐..б?|шЯ"
00A0  00B80102 E8B30072 198BFBB9 0B00BED6  7DF3A675 0D8D7F20 BEE17DB9 0B00F3A6  ".┐..ш|.r.Л√╣..┘╓}≤жu.Н. ┘с}╣..≤ж"
00C0  7418BE77 7DE86A00 32E4CD16 5E1F8F04  8F4402CD 19BEC07D EBEBA11C 0533D2F7  "t.┘w}шj.2ф═.^.П.ПD.═.┘└}ыыб..3╥≈"
00E0  360B7CFE C0A23C7C A1377CA3 3D7CBB00  07A1377C E84900A1 187C2A06 3B7C4038  "6.|■└в<|б7|г=|┐..б7|шI.б.|*.;|@8"
0100  063C7C73 03A03C7C 50E84E00 5872C628  063C7C74 0C010637 7CF7260B 7C03D8EB  ".<|s.а<|PшN.Xr╞(.<|t...7|≈&.|.╪ы"
0120  D08A2E15 7C8A16FD 7D8B1E3D 7CEA0000  7000AC0A C07422B4 0EBB0700 CD10EBF2  "╨К..|К.²}Л.=|ъ..p.м.└t"┤.┐..═.ы≥"
0140  33D2F736 187CFEC2 88163B7C 33D2F736  1A7C8816 2A7CA339 7CC3B402 8B16397C  "3╥≈6.|■┬И.;|3╥≈6.|И.*|г9|├┤.Л.9|"
0160  B106D2E6 0A363B7C 8BCA86E9 8A16FD7D  8A362A7C CD13C30D 0A4E6F6E 2D537973  "▓.╥ц.6;|Л╩ЖщК.²}К6*|═.├..Non-Sys"
0180  74656D20 6469736B 206F7220 6469736B  20657272 6F720D0A 5265706C 61636520  "tem disk or disk error..Replace "
01A0  616E6420 73747269 6B652061 6E79206B  65792077 68656E20 72656164 790D0A00  "and strike any key when ready..."
01C0  0D0A4469 736B2042 6F6F7420 6661696C  7572650D 0A00494F 20202020 20205359  "..Disk Boot failure...IO      SY"
01E0  534D5344 4F532020 20535953 00000000  00000000 00000000 00000000 000055AA  "SMSDOS   SYS..................Uк"
```

Дамп сектора, содержащего усеченный Boot-сектор

```
0000  EB289049 424D2050 4E434900 02040100  0200023E 9FF82800 22000600 22000000  "ы(PIBM PNCI........>Я°(."..."..."
0020  00000000 00000000 0000FA33 C08ED0BC  F07BFBB8 C0078ED8 BE5B0090 FCAC0AC0  "..........3└O╨╝≡{√╕└.О╪┘[.P"м.└"
0040  740B56B4 0EBB0700 CD105EEB F032E4CD  16B40FCD 1032E4CD 10CD190D 0A0D0A0D  "t.V┤.┐..═.^ы≡2ф═.┤.═.2ф═.═......"
0060  0A0D0A0D 0A0D0A0D 0A0D0A20 20202054  68697320 6469736B 20697320 6E6F7420  "...........    This disk is not "
0080  626F6F74 61626C65 0D0A0D0A 20496620  796F7520 77697368 20746F20 6D616B65  "bootable.... If you wish to make"
00A0  20697420 626F6F74 61626C65 2C0D0A72  756E2074 68652044 4F532070 726F6772  " it bootable,..run the DOS progr"
00C0  616D2053 59532061 66746572 20746865  0D0A2020 20202073 79737465 6D206861  "am SYS after the..     system ha"
00E0  73206265 656E206C 6F616465 640D0A0D  0A506C65 61736520 696E7365 72742061  "s been loaded....Please insert a"
0100  20444F53 20646973 6B657474 6520696E  746F0D0A 20746865 20647269 76652061  " DOS diskette into.. the drive a"
0120  6E642073 7472696B 6520616E 79206B65  792E2E2E 00000000 00000000 00000000  "nd strike any key..............."
0140  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0160  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
0180  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01C0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 000055AA  "..............................Uк"
```

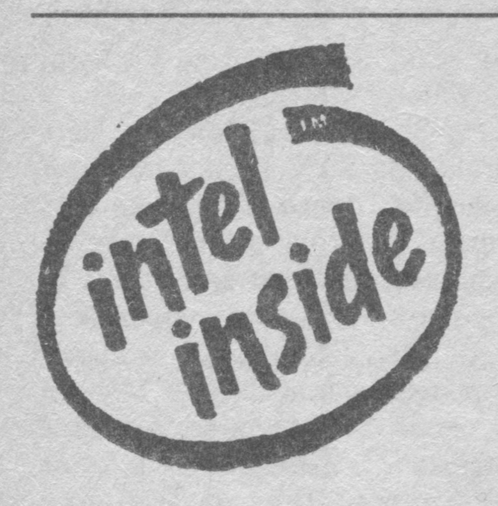
Дамп сектора, зараженного вирусом "Stone"

```
0000  EA0500C0 07E99900 0043A300 F0E40040  9F007C00 001E5080 FC027217 80FC0473  "ъ..└.щЩ..Cг.≡ф.@Я.|...PА".r.А".s"
0020  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  "..╤u.3└O>a?.и.u.ш..X......SQR.VW"
0040  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  "┘..┐....┐..3╔Л╤АЬ....s.3└Ь......"
0060  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  "NuрыБР3÷┐.."..н;.u.н;E.t!┐..┐..▌"
0080  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  ".╣.Ь....r.┐..3█.3╤Ь......_^.ZY["
00A0  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  "├3└O> O╨╝.|√бL.r.|бN.r.|б..HHr.."
00C0  B106D3E0 8EC0A30F 7CB81500 A34C008C  064E00B9 B8010E1F 33F68BFE FCF3A42E  "▓.╙рО└г.|╕..гL.М.N.╣┐...3÷Л■"≤д."
00E0  FF2E0D00 B80000CD 1333C08E C0B80102  BB007C2E 803E0800 00740BB9 0700BA80  "....┐..═.3└O└┐..┐.|.А>.....t.╣..║А"
0100  00CD13EB 4990B903 00BA0001 CD13723E  26F6066C 04077512 BE89010E 1FAC0AC0  ".═.ыIР╣..║..═.r>&÷.l..u.┘Й...м.└"
0120  7408B40E B700CD10 EBF30E07 B80102BB  0002B101 BA8000CD 1372130E 1FBE0002  "t.┤.┐.═.ы≤..┐..┐..▓.║А.═.r...┘.."
0140  BF0000AD 3B057511 AD3B4502 750B2EC6  06080000 2EFF2E11 002EC606 080002B8  "┐..н;.u.н;E.u..╞...........╞....┐"
0160  0103BB00 02B90700 BA8000CD 1372DF0E  1F0E07BE BE03BFBE 01B94202 F3A4B801  "..┐..╣..║А.═.r▀.....┘┘.┐┘.╣B.≤д┐."
0180  0333DBFE C1CD13EB C507596F 75722050  43206973 206E6F77 2053746F 6E656421  ".3█■┴═.ы┼.Your PC is now Stoned!"
01A0  070D0A0A 004C4547 414C4953 45204D41  52494A55 414E4121 00000000 00000000  ".....LEGALISE MARIJUANA!........"
01C0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
01E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  "................................"
```

Дамп файла, пораженного вирусом "Vienna-648"

```
С адреса 0000 по адрес 0002 (3 байта)   -- команда перехода на тело вируса.
С адреса 0003 по адрес 0CA9             -- файл.
С адреса 0CAA по адрес 0F31             -- вирус.

0000  E9A70C89 160D03B4 30CD218B 2E02008B  1E2C008E DAA3B70B 8C06B50B 891EB10B   "щз.Й...┤0═!Л...Л.,.О┌г╖.М.╡.Й.▒."
....  ........ ........ ........ ........  ........ ........ ........ ........   ................................
09E0  0A0BC074 370BDB74 2D83EB04 8B0F498B  D083C205 83E2FE83 FA087303 BA08003B   "..└t7.█t-Гы.Л.IЛ╨Г┬.Гт■Г·.s.║..;"
0A00  CA720C77 0583C304 EB08E887 FFEB03E8  4FFF8BC3 EB0D50E8 65FEEB06 53E890FD   "╩r.w.Г├.ы.шЗ.ы.шO.Л├ы.Pшe■ы.SшР²"
0A20  33C05B5D 5F5EC300 00000000 00000000  00021F05 00000000 00000000 00000000   "3└[]_^├........................."
0A40  00000000 54757262 6F20432B 2B202D20  436F7079 72696768 74203139 39302042   "....Turbo C++ - Copyright 1990 B"
0A60  6F726C61 6E642049 6E746C2E 00446976  69646520 6572726F 720D0A41 626E6F72   "orland Intl..Divide error..Abnor"
0A80  6D616C20 70726F67 72616D20 7465726D  696E6174 696F6E0D 0A000000 00000000   "mal program termination........."
0AA0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 000000EA   "...............................ъ"
0AC0  0DEA0D00 00000000 00000000 00000000  00003A03 3A033A03 00000902 00000000   ".ъ................:.:.:........."
0AE0  00000000 0000D80B 00000A02 01000000  00000000 0000E80B 00000202 02000000   ".......>..............ш........."
0B00  00000000 0000F80B 00004302 03000000  00000000 0000080C 00004202 04000000   "......°...C...............B....."
0B20  00000000 0000180C 00000000 FF000000  00000000 0000280C 00000000 FF000000   "......................(........."
0B40  00000000 0000380C 00000000 FF000000  00000000 0000480C 00000000 FF000000   "......8...............H........."
0B60  00000000 0000580C 00000000 FF000000  00000000 0000680C 00000000 FF000000   "......X...............h........."
0B80  00000000 0000780C 00000000 FF000000  00000000 0000880C 00000000 FF000000   "......x...............И........."
0BA0  00000000 0000980C 00000000 FF000000  00000000 0000A80C 00000000 FF000000   "......Ш...............и........."
0BC0  00000000 0000B80C 00000000 FF000000  00000000 0000C80C 00000000 FF000000   "......╕...............╚........."
0BE0  00000000 0000D80C 00000000 FF000000  00000000 0000E80C 00000000 FF000000   "......>...............ш........."
0C00  00000000 0000F80C 00000000 FF000000  00000000 0000080D 01200220 022004A0   "......°.................. . . .а"
0C20  02A0FFFF FFFFFFFF FFFFFFFF FFFFFFFF  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF   ".а.............................."
0C40  00000000 00000010 00000013 02020405  06080808 14150513 FF160511 02FFFFFF   "................................"
0C60  FFFFFFFF FFFFFFFF FFFF0505 FFFFFFFF  FFFFFFFF FFFFFFFF FFFFFFFF 0FFF2302   "...#............................"
0C80  FF0FFFFF FFFF13FF FF020205 0F02FFFF  FF13FFFF FFFFFFFF FFFF23FF FFFFFF23   "..........................#....#"
0CA0  FF13FF00 00000000 000051BA A30FFC8B  F281C60A 00BF0001 B90300F3 A48BF2B4   "..........Q║г.ⁿЛ≥Б╞..┐..╣..≤дЛ≥┤"
0CC0  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX   "0═!<.u.щ╟..┤/═!ЙЬ..МД...▐_.Р.╓┤."
0CE0  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX   "═!.VO.,.┐..^VБ╞..м╢.А≥о╢..моuэт."
0D00  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX   "^.Й╝..Л■Б╟..Л▐Б╞..Л■ы:Г╝...u.щс."
0D20  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX   ".V&O.,.Л■&Л╡..Б╟..м<;t.<.t.кы⌠╛."
0D40  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX   ".[.Й╕..А}\t..▒\кЙ┐..Л≤Б╞..╢..≤дЛ"
0D60  F3B44EBA 1F009003 D6B90300 CD21EB04  B44FCD21 7302EB9E 8B847500 241F3C1F   "≤┤N║..Р.╓╣..═!ы.┤O═!s.ыЮЛДu.$.<."
0D80  74EE81BC 790000FA 77E683BC 79000A72  DF8BBC18 005681C6 7D00ACAA 3C0075FA   "tюБ╝y..·wцГ╝y..r▀Л╝..VБ╞}.мк<.u·"
0DA0  5EB80043 BA1F0090 03D6CD21 898C0800  B8014381 E1FEFFBA 1F009003 D6CD21B8   "^╕.C║..Р.╓═!ЙМ..╕.CБс■.║..Р.╓═!╕"
0DC0  023DBA1F 009003D6 CD217303 E9A5008B  D8B80057 CD21898C 04008994 0600B42C   ".=║..Р.╓═!s.ще.Л╪╕.W═!ЙМ..ЙФ..┤,"
0DE0  CD2180E6 077510B4 40B90500 8BD681C2  8A00CD21 EB6590B4 3FB90300 BA0A0090   "═!Ац.u.┤@╣..Л╓Б┬К.═!ыeР┤?╣..║..Р"
0E00  03D6CD21 72553D03 007550B8 0242B900  00BA0000 CD217243 8BC82D03 0089840E   ".╓═!rU=..uP╕.B╣..║..═!rCЛ╚-..ЙД."
0E20  0081C1F9 028BFE81 EFF70189 0DB440B9  88028BD6 81EAF901 CD21721F 3D880275   ".Б┴∙.Л■Бя≈.Й.┤@╣И.Л╓Бъ∙.═!r.=И.u"
0E40  1AB80042 B90000BA 0000CD21 720DB440  B903008B D681C20D 00CD218B 9406008B   ".╕.B╣..║..═!r.┤@╣..Л╓Б┬..═!ЛФ..Л"
0E60  8C040081 E1E0FF81 C91F00B8 0157CD21  B43ECD21 B801438B BC0800BA 1F009003   "М..Бср.Б╔..╕.W═!┤>═!╕.CЛ╝..║..Р."
0E80  D6CD211E B41A8B94 00008E9C 0200CD21  1F5933C0 33DB33D2 33F6BF00 015733FF   "╓═!.┤.ЛФ..ОЬ..═!.Y3└3█3╥3÷┐..W3."
0EA0  C2FFFF80 004018D1 54B81620 008CCA2E  E9A70C2A 2E434F4D 00280018 04504154   "┬..А.@.╤T╕. .М╩.щз.*.COM.(...PAT"
0EC0  483D5445 4D50322E 434F4D00 4D00454D  2E434F4D 0000434F 4D002020 20202020   "H=TEMP2.COM.M.EM.COM..COM.      "
0EE0  20202020 20202020 20202020 20202020  20202020 20202020 20202020 20202020   "                                "
0F00  2020043F 3F3F3F3F 3F3F3F43 4F4D0302  00E71300 00000020 D154B816 AA0C0000   "  .????????COM...ч..... ╤T╕.к..."
0F20  54454D50 322E434F 4D004D00 00EA0B02  1358                                  "TEMP2.COM.M..ъ...X"
```



Интелвью & IntelView.

Как делать бизнес. Мысли серьезного специалиста

КомпьютерПресс: Дмитрий, расскажите, пожалуйста, о том, как прошел первый год работы в СССР и каковы его итоги для Intel.

Дмитрий Ротов: Не было ни сотрудников, ни денег для того, чтобы организовать работу здесь, не было ни команды с Запада, чтобы заниматься этой работой, ни решения о ее создании. В общем, все прелести жизни. В июне я представил доклад своему правительству и руководству фирмы Intel. И я говорил, что я не думаю, что это будет выгодно нам, не будет выгодно год, или два года, или три, но через пять лет, уж точно, будет налажена настоящая работа. Хотя лично я начал работать в России уже тогда. Нам надо было создать здесь команду, заниматься обучением, найти дистрибьюторов и создать дистрибьюторскую сеть.

Чтобы осуществить все это, надо было подобрать человека, которому придется жить или в Европе, или в Москве. И мне сказали: "О'Кей, Дмитрий, ты и будешь этим человеком". И тогда же я пригласил Стива Чейза (Stephen Chase). Он стал нашим первым сотрудником здесь, это наш менеджер, он возглавляет все работы в России. Он проработал 12 лет на фирме Intel в центральном регионе. Он занимается продажей, он эксперт, он знает эту область, и для нас важно, чтобы именно такой опытный менеджер работал на этом рынке. Наша структура такова: все, кто отвечает за продажи, работают с ним, с нашим менеджером, он их руководитель, он сконцентрирован на этих вопросах. Все остальные вопросы: маркетинг, сотрудничество, обучение, транспорт — решаем я и Марат*. Короче говоря, мы пытаемся обеспечить все условия работы для группы, занимающейся сбытом. Ведь в России работать сложно, сейчас меняется вся система, постоянно нужно что-то добывать и улаживать, искать какие-то бумаги, конечно, нам не до продаж.

Сперва мы открыли здесь бюро вместе с фирмой Microsoft. И только сейчас подписали контракт и открыли новый офис — 370 м^2, у нас есть 9 или 10 сотрудников здесь, в России, которые работают в Москве и Санкт-Петербурге. Теперь есть сотрудники и в Киеве. Когда в прошлом году мы встретились с Маратом, и он стал нашим представителем здесь, у нас почти не было здесь продаж. Тогда мы вложили настоящие деньги, нашли дистрибьюторов...

У нас есть сильные хорошие люди, и они хотят быстро продвигаться на рынок. Для этого они усиленно работают. Довольно неожиданно, но наши объемы продаж оказались в четыре раза больше, чем мы планировали на этот год.

КП: Это по сравнению с тем, что планировалось на первый год работы?

Д.Р.: Да, в четыре раза больше. Но это был уже второй год, первый год был неполный. Он начался в октябре. Тогда здесь еще почти никто не работал. Были только Марат, Игорь** и я. Я живу сейчас в Германии, это для того, чтобы быть ближе к России. До этого мне приходилось ездить по два раза в месяц туда-сюда в Калифорнию и обратно. Это было очень тяжело, я почти умирал. И поэтому сейчас я переехал в Германию. Сейчас я бываю неделю здесь, неделю там.

Так вот, в первый год у нас было четыре дистрибьютора, но трое из них были неработающими. После этого мы решили работать только с серьезными дистрибьюторами. Мы опубликовали несколько объявлений, в которых говорили, что нам нужно: ну, там наличие квалифицированных серьезных людей, 50 000 долларов для первого вклада, желание продавать новую продукцию. Дистрибьютор работает в двух разных направлениях: во-первых, он продвигает новую технологию, то есть он должен иметь хороших специалистов в этой области, но, во-вторых, он работает как банк (с деньгами): покупатель дает ему заказ, он должен дать заказ нам, и между двумя заказами он должен что-то делать. Либо дать кредит покупателю (и нам), либо принять деньги у покупателя и передать нам, то есть это работа, как у банка. Только в этом случае он может брать большие заказы и делать нормальный бизнес, собирая зака-

*Марат Ишуев — менеджер фирмы Intel в бывшем СССР.

**Игорь Илларионов — переводчик и помощник по маркетингу в московском офисе фирмы Intel.

зы от клиентов. И я рад, что мы нашли несколько очень хороших дистрибьюторов.

КП: А кто это?

Д.Р.: Сейчас у нас их шесть: Центр ПЭВМ "Техно", фирма "Стинс Коман", Merisel CAT, ComputerLand, в Киеве — Квазар Микро, а в Санкт-Петербурге — NEVVIS.

КП: Интересно, откликнулись ли какие-то люди на те ваши объявления, что нужны дистрибьюторы?

Д.Р.: Очень многие. Конечно, у всех интерес был разный. Мы очень часто приглашаем людей к сотрудничеству с Intel, но при этом хотим, чтобы это было выгодно не только нам, но и нашим партнерам. Это серьезная работа. Сейчас еще один наш интерес состоит в том, чтобы иметь свободных сотрудников и развивать математику. Для этого нам также нужны люди, отлично владеющие этой областью, имеющие особый дар. Сейчас у нас есть несколько таких человек. Правда, желающих слишком много: каждый, у кого есть персональный компьютер, говорит, что именно он знает, как сделать самый новый, самый хороший процессор.

И, конечно, каждый из них хочет работать с Intel. Но, естественно, очень трудно встретить людей уровня, скажем, Келдыша, способных делать серьезные работы. Но все-таки, мы будем развивать это направление. В прошлом году мы уже создали настоящую команду здесь и на Западе. И, в итоге, приняли решение делать все здесь, не на Западе. Это выгодно и правильно стратегически. Я очень доволен нашими сотрудниками здесь, тем, что они могут делать. Они очень квалифицированны. Они были на Западе очень много раз, были в Нью-Йорке, Париже и т.д.

Итак, сейчас мы открыли настоящее бюро, мы делаем здесь настоящий бизнес, у нас настоящие дистрибьюторы, мы подписали контракт, чтобы делать программное обеспечение здесь. Я думаю, что мы хорошо начинали, первый год был успешным. И сейчас начался второй год, и нам еще нужны новые дистрибьюторы. На будущий год мы хотим открыть несколько бюро в разных городах, это будут наши представительства. Сейчас они работают в нескольких городах Сибири, Санкт-Петербурге, а также "за границей", в Киеве, в Минске, в некоторых городах Прибалтики, кроме того, у нас работают представители этих республик здесь: очень интернациональная команда.

Мы сделаем передвижную выставку, и будем, таким образом, проводить серии семинаров в Москве, Киеве, Минске, Питере, на Волге, на Транссибирской магистрали, в Прибалтике. Дело в том, что там более 300 заводов. Мы хотим наладить связи с российскими заводами, занимающимися экспортом и имеющими деньги. Мы посылаем выставку, наш стенд, специалиста — все это мероприятие занимает три месяца. Это у правительства нет денег, а у многих заводов есть большие деньги. Заводы продают на экспорт станки, самолеты, автобусы, двигатели, многое другое, а всем нужны процессоры. Сперва небольшое число.

КП: Да, но не многим нужны микропроцессоры.

Д.Р.: Вот пример: завод хочет наладить выпуск новой серии автомобилей, и ему нужен микроконтроллер для управления двигателем, тормозами, подвеской. Это и экология. Все это нужно, чтобы выйти на уровень западных стандартов, для развития экспорта. По затратам это ничто, это 20-30 долларов, но это ключ, ключ к успеху экспорта. И я, как человек, считаю, что это хорошее дело — помогать таким заводам — хорошее для бизнеса здесь, хорошее для нас, хорошее для людей.

КП: Судя по всему, дела у Intel даже лучше, чем ожидалось. Это хорошо, хорошо если и дальше они будут идти лучше и лучше. Но наш следующий вопрос скорее о планах: что Вы думаете о соотношении основных фирм, которые сейчас делают процессоры? Как бы Вы определили позицию таких фирм, как Sun, Hewlett-Packard, Intel, Motorola, MIPS?

Д.Р.: Здесь есть Intel и есть Motorola. На 90% рынка работает фирма Intel, около 8% рынка есть у фирмы Motorola, на все остальные фирмы приходится лишь 2%. В прошлом году объем продаж у фирмы Intel составил 20 миллионов процессоров, у фирмы Motorola — около 3 миллионов, а у всех остальных вместе — около 400 000 штук. Даже на этой выставке вы можете видеть то количество фирм, которое выпускает свои компьютеры на базе процессоров Intel.

КП: Практически все.

Д.Р.: Практически все. И я не имею ничего против MIPS, SUN или Hewlett-Packard. Это приличные фирмы, хорошие умные люди, хорошие архитектуры, но рынок очень широк. Они говорят себе: "Я вижу, что рынок очень широк, и я найду себе узкий сектор рынка, и я займу его, и буду делать процессор для него." Нам это не интересно. Нас интересует широкий рынок и мы делаем процессоры для него.

КП: А что можно сказать о направлениях каждой из этих фирм? Они все к чему-то стремятся. Intel стремится завоевать максимум рынка и дать широкие возможности пользователю.

Д.Р.: Вот хороший пример. Девять месяцев назад на выставке COMDEX мы показали 100 МГц процессор. И нам было бы очень просто выпускать рабочие станции, используя этот 100 МГц процессор. Но это было бы очень узким интересом. А нужно видеть широкие задачи, широкий рынок, широкие проблемы. Развивать надо то, что необходимо широкому кругу покупателей. У тех трех тысяч фирм, что делают компьютеры, есть главная проблема: им трудно делать компьютер с частотой более 33 МГц. Только несколько из них могут делать 50 МГц. А компьютер на 66 МГц уже поднимает проблемы аналоговой техники.

Мы никогда себе не ставили задачи создать такой процессор, ко-

торый мы и только мы сможем использовать в своей системе. Наша задача помогать трем тысячам фирм и их покупателям. Как помогать им? Эта задача гораздо труднее. Поэтому мы внедрили новую технологию, которая позволяет ставить новый чип на старую машину, получая новую систему.

КП: Это по сути дела и есть Overdrive?

Д.Р.: Это один из примеров Overdrive, будет несколько таких. Суть Overdrive поясняет простой пример: пользователь просто ставит новый чип рядом со старым, не вынимая его. Многие боятся снимать процессор с системы. Просто-напросто.

КП: То есть получается, что старый процессор практически не используется?

Д.Р.: Иногда да.

КП: То есть он просто стоит на плате как лишний элемент?

Д.Р.: Есть разные варианты, и это лишь один из них. В некоторых случаях можно снимать старый чип и ставить новый. Есть специальное гнездо, чтобы это было удобно делать. У меня есть компьютер notebook-Toshiba. Сначала там стоял процессор 486SX, потом я купил и вставил новый процессор 486DX, теперь я поставил туда новый процессор 486DX2. Но многие не хотят делать так, они хотят оставить и старый процессор. Это — вопрос вкуса.

КП: Вопрос о флэш-памяти. Тут опять можно аплодировать Intel, я слышал, что появилась карта флэш-памяти емкостью 8 Мбайт.

Д.Р.: Да, есть такая, и скоро будет карта 20 Мбайт с флэш-памятью.

КП: Какова роль фирмы Sharp в этих совместных исследованиях? Почему именно Sharp?

Д.Р.: Мы очень много работали с Sharp. Мы верим, что Sharp очень хорошо знает рынок потребительских товаров. И они работают в области компьютеров. И мы думаем, что сотрудничество с Sharp поможет нам выйти с флэш-памятью на очень большой рынок. Причем, гораздо более широкий, чем рынок компьютеров (хотя компьютеры — это тоже большой рынок), прежде всего это "умные" кредитные карточки, "умные" телефоны,— это огромный рынок. И Sharp занимается этой областью, и очень хорошо занимается. И у них есть технология производства жидкокристаллических экранов, она очень хороша. Я видел эту продукцию на выставке в Ганновере, изумительная разрешающая способность, изображение. Я никогда не видел такого ясного и чистого изображения. Они показывали там проектор с высоким разрешением. Когда я его увидел, то подумал, что это кино — чистое изображение, как живое.

КП: Но вернемся к флэш. Понятно, что увеличение количества мегабайт на плате — это хорошо, но наверное, фирма работает и над увеличением скорости записи?

Д.Р.: Да, и это тоже. Вообще говоря, флэш-память — это один из наиболее сложных видов продукции, разработанных фирмой Intel. Необходимо было решить три очень важных задачи, чтобы обеспечить широкий рынок для этих изделий. Во-первых, повысить плотность записи информации, то есть, по сути, увеличить емкость микросхем, во-вторых, сделать возможным стирание не всей записанной в чипе информации, а лишь части, и, в-третьих, снизить цену подобных изделий до уровня цен динамических ОЗУ.

КП: Если увеличивается скорость записи, значит винчестеры умрут?

Д.Р.: У нас есть фирма Conner, одна из самых сильных среди выпускающих маленькие винчестеры больших объемов. Есть направление деятельности — выпускать совместно с ними накопители таких же размеров на флэш-памяти.

КП: То есть, фактически, это будет устройство, которое заменит винчестер компьютера?

Д.Р.: Может быть. И можно будет работать гораздо лучше. Здесь на выставке можно увидеть нашу новую систему. Это очень быстрая система, потому что у нее очень хорошая внутренняя организация и очень быстрый винчестер. В ней использован интерфейс SCSI-2, то есть в передаче данных шина не участвует, данные попадают прямо с диска в процессор. И поэтому увеличивается скорость работы системы. Почему чувствуется, что система так быстро работает? Потому что в ней очень быстрые диски. Теперь можно предположить, как работала бы система, если вместо диска была бы флэш-память.

КП: Видимо, не за горами появление компьютеров вообще без механики, то есть винчестеры отпадают, процессоры не греются, следовательно не нужен вентилятор, "кредитная карточка" на 100 Мбайт вместо флоппи-диска. Получается, что в компьютере ничего не движется, то есть это потрясающе надежная система, которая может появиться буквально через несколько лет.

Д.Р.: Да, что-то похожее вы можете уже увидеть здесь. В нашей системе уже всего одна плата, на которой расположено все, но пока есть винчестер и флоппи.

И тут мы пошли смотреть на компьютер Дмитрия, кстати, сделанный фирмой Intel. Компьютер действительно оказался что надо — он делал все, что угодно, и очень быстро. После этого мы тепло попрощались с Дмитрием, выразив надежду, что эта встреча не была последней. Так оно и оказалось...

А.Борзенко,
И.Вязаничев



Наверное, все в нашей стране если не работали с ним, то по крайней мере слышали о текстовом редакторе Лексикон. Наверное, многие знают о существовании интегрированного пакета Мастер. С этого номера мы начинаем печатать серию статей о вышеуказанных продуктах, а также о их разработчиках.

СП "Микроинформ". Верной дорогой идете, господа-товарищи!

Начало

23 июня 1988 года Минфином СССР было зарегистрировано 60-е совместное предприятие. Учредителем СП с советской стороны являлся Центральный межведомственный институт повышения квалификации руководящих работников и специалистов строительства (51% из миллиона уставного капитала), с зарубежной стороны — венгерское предприятие "Телефондьяр". Новое предприятие получило название "Микроинформ". В штате тогда было 10 человек, но и они с трудом размещались в нескольких комнатках старого дома № 3 на Кожевнической улице.

Через три года "Микроинформ" справил новоселье, вложив в ремонт и реставрацию полуразрушенного здания 3 миллиона инвалютных рублей. Теперь фирма размещается в прекрасном четырехэтажном особняке по улице Островского, 44. Очевидно, заключенный летом 88-го брак по расчету оказался удачным!

Компьютеры

Вопрос: На чем можно было заработать за 3 года такие деньги?

Ответ: Конечно, на компьютерах! Но... не все так просто (а иначе и писать было бы не о чем). Послушайте, что говорит генеральный директор "Микроинформа" Борис Михайлович Фридман о своей фирме: "Мы ставим перед собой цель обеспечить страну продукцией, на равных конкурирующей с зарубежной. Для производства компьютеров есть и исходная база, и комплектующие, и технология — нужен лишь механизм, структура для реализации имеющихся возможностей. Мы не готовы пока к организации в стране даже сборочного производства. Сегодня выгоднее вложить средства в рациональное совместное использование свободных мощностей высокотехнологичных производств, имеющихся на Западе. Компоненты наших полностью совместимых с IBM AT компьютеров приобретаются у ведущих западных фирм, а для сборки машин мы арендуем технологическую линию у германской фирмы Soft-tronik, являющейся нашим партнером. От такой организации производства выигрывают все, и прежде всего — наши заказчики".

А заказать в "Микроинформе" можно много хорошего и разного. Имея налаженные связи с зарубежными партнерами, "Микроинформ" может приобрести и поставить заказчику нужную продукцию и других фирм — например, полностраничные мониторы вертикального формата и другую специальную технику для современных компьютерных издательских систем, сетевое оборудование.

Немаловажно наличие в "Микроинформе" технического центра, занимающегося установкой, техническим обслуживанием, гарантийным и послегарантий-



С этого все начиналось...

ным ремонтом аппаратуры. В случае технических неприятностей клиентам фирмы не надо искать, к кому обратиться за помощью.

Программное обеспечение

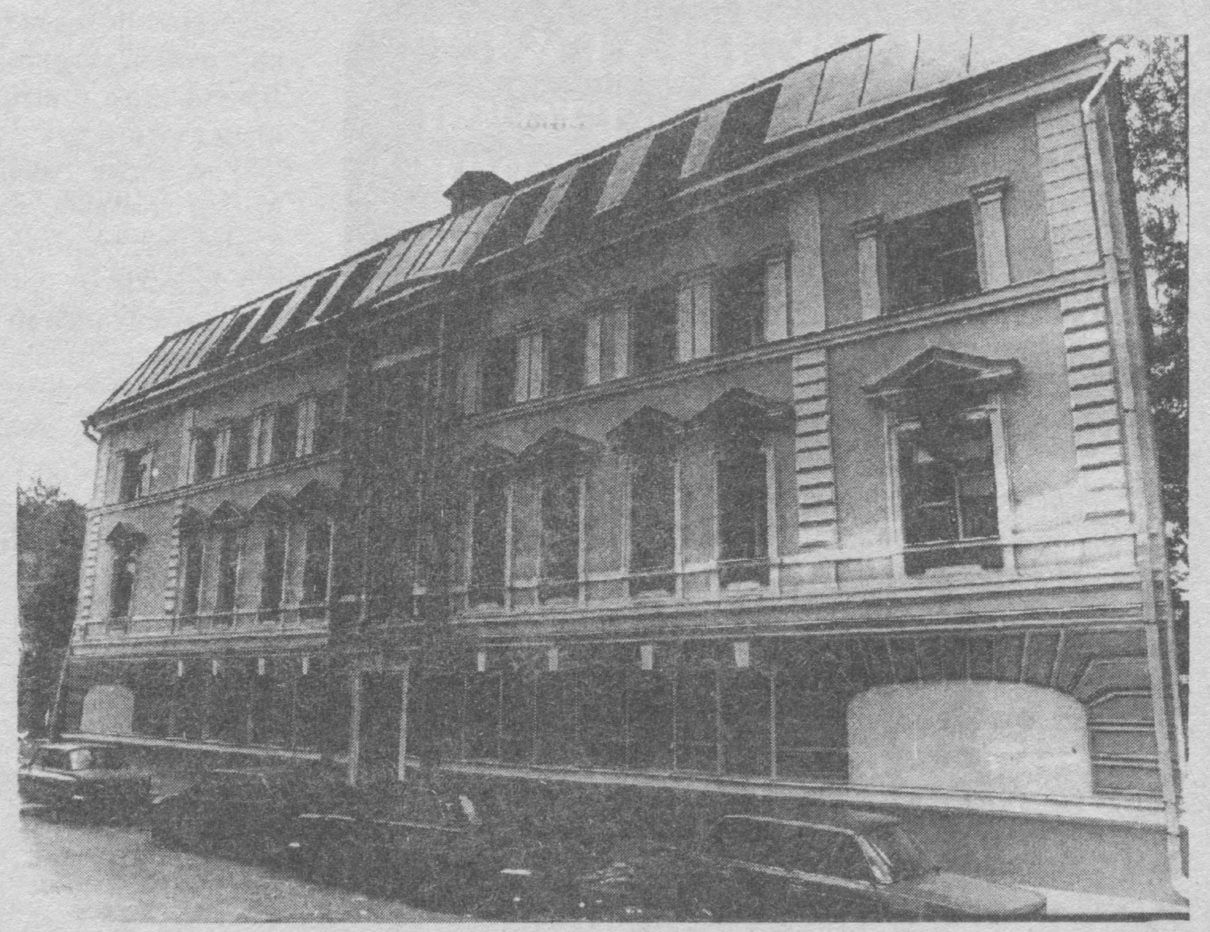
Как правило, "Микроинформ" не ограничивается поставкой "голых" машин — заказчикам предлагаются готовые к работе аппаратно-программные комплексы. "Микроинформ" является официальным дистрибьютором продукции фирм Computer Associates и Rank Xerox, осуществляя поставку и сопровождение в СССР ряда программ. Xerox славен не только как родоначальник копировальных аппаратов, сделавших название фирмы нарицательным, но и разработкой издательской системы Xerox Ventura Publisher. Предлагаемая на нашем рынке версия Ventura Publisher полностью русифицирована и снабжена фирменной документацией на русском языке.

Безусловно, заслуживают внимания и продукты Computer Associates: система управления проектами Super Project Expert (инструмент для руководителя крупной организации) и всем известный табличный процессор SuperCalc.

Многочисленные пользователи SuperCalc хорошо знают как его широкие возможности, так и главный недостаток: на диаграммах и графиках не удается сделать подписи и обозначения по-русски. Это так, если только... не приобрести пакет SuperCalc *5* в "Микроинформе", для которого фирма предоставляет программную русификацию (HELP, экранные и принтерные графические шрифты), а также переведенный на русский язык комплект фирменной документации. Интересующихся возможностями SuperCalc *5* отошлем к статье О.Ф.Литвина, опубликованной в КомпьютерПресс № 4´91.

Мастер и Лексикон

Пожалуй, наиболее интересными аспектами деятельности "Микроинформа" являются собственные программные разработки. Сочетание под крышей



...зато теперь...

одной небольшой фирмы поставок компьютеров и создания программного обеспечения вряд ли оправдало бы себя в условиях развитого рынка программ и техники. Однако в наших условиях для заказчика надежнее получить технику и программное обеспечение (а также обучение и сервис) из одних рук, поэтому именно такое полное обслуживание стало основным принципом работы "Микроинформа".

Основные работы "Микроинформа" идут в русле технологии программного комплекса Мастер — полностью оригинальной, лицензионно чистой интегрированной системы. Частью этой системы и одновременно самостоятельно развивающимся продуктом является известный едва ли не всем советским пользователям IBM-совместимых машин — редактор текстов Лексикон.

Мастер и Лексикон — детища Е.Н.Веселова, ставшего в этом году техническим директором "Микроинформа".

Чем примечателен Лексикон? Ведь существуют текстовые процессоры гораздо более мощные, с богатыми возможностями оформления документов, включением в текст графических иллюстраций, автоматизированной работой с внутренней структурой текста (главы и разделы, сноски, ссылки), встроенными картотеками, калькуляторами и т.д.

Тем не менее большинство пользователей охотно работают с Лексиконом — копия той или иной его версии найдется у нас в стране, наверное, на каждом втором компьютере. Это объясняется тем, что:

- Лексикон легко освоить, начать работу можно практически без всякой подготовки, постепенно накапливая опыт и навыки;
- при внешней простоте работы он располагает всеми необходимыми функциями обработки текстов;
- для работы не надо знать ни слова по-английски; с другой стороны, можно работать с англоязычными и смешанными текстами;
- для печати годится практически любой принтер: как матричный, так и лазерный.

Возможно, и на вашем компьютере есть Лексикон версии 6.67 или даже более ранний, созданный Е.Н.Веселовым еще в ВЦ АН СССР. Версия 6.67 была последней, распространявшейся на некоммерческой основе. Именно она описана в книге В.Э.Фигурнова "IBM PC для пользователя", изданной в 1991 году издательствами "Финансы и статистика" и КомпьютерПресс. Но время не стоит на месте. Успех Лексикона побудил разработчиков создать на основе прежнего продукта практически новую программу, получившую название Лексикон.

Загляните в один из появившихся наконец и у нас в крупных книжных магазинах отделов по продаже программного обеспечения. Типичная картина: на полках теснятся роскошные коробки западных фирм, а внизу, под стеклом прилавка — россыпь дискеток производства "собственных Платонов и быстрых разумом Ньютонов", снабженных вышедшими из-под 9 иголок листочками с рекламой и пояснениями.

Благодаря сотрудничеству с Казанским комбинатом программных средств удалось наконец создать не просто очередную версию программы, а настоящий Программный Продукт со всеми его атрибутами — руководством пользователя, паспортом и регистрационной карточкой, этикетками, яркой упаковкой. Коробку с Лексиконом и вышедшим вслед за ним Мастером приятно взять в руки и не стыдно поставить на витрину.

Увесистые глянцевые кирпичи Мастера стали важным и довольно рискованным шагом "Микроинформа" к цивилизованному рынку: Мастер начал поставляться без защиты от копирования. Так как полный комплект инструментальной системы ориентирован в основном на разработчиков, есть надежда, что проторенной дорожке DISKCOPY коллеги предпочтут тернистый легальный путь, обещающий зато поддержку фирмы.

К сожалению, Лексикон по-прежнему выпускается лишь в защищенном виде — слишком прост он в освоении и слишком широк круг его пользователей, чтобы деловые или этические соображения могли препятствовать пиратскому размножению. А ведь сколько неудобства и головной боли приносят и программистам, и пользователям "ключевые диски" и "инсталляции"...

Но не оскудела еще талантами наша земля! На всякую защиту находятся умельцы, способные ее взломать и промышляющие экскурсиями в чужие программы (карманы, квартиры — по способностям). К сожалению, законодательство пока благоволит к компьютерным карманникам, и по стране бродят беспри-

зорные копии программ с испорченной системой защиты. Авторы же вместо совершенствования программ вынуждены заботиться о все более хитроумных запорах — в противном случае им вскоре пришлось бы переквалифицироваться в дворники.

Снятие защиты не всегда проходит бесследно. Если однажды при работе с такой копией во время записи набранного текста на диск машина "зависнет" и результаты многочасового труда пойдут прахом — не вините вирусы, программу или технику, виновата лишь ваша непредусмотрительность. Так что, если на компьютере есть старая или нелегальная версия Лексикона и работа с этой программой вам нравится — соберитесь с духом и уничтожьте ее! Пользуйтесь только легальными копиями программы!

Даже для самого неполного рассказа об интегрированной системе Мастер рамки небольшой статьи тесноваты, о нем, как и о Лексиконе, будет гораздо более подробно рассказано в ближайших номерах КомпьютерПресс, сейчас же лишь краткое знакомство.

Мастер предоставляет пользователю возможности аналогичного Лексикону многооконного текстового процессора, системы взаимосвязанных между собой электронных таблиц, средства визуализации и создания пользовательского интерфейса оконного типа, и, наконец, мощные средства для работы с базами данных и язык программирования высокого уровня Мастер. Программируя на Мастере, с СУБД и информационной средой можно делать буквально все что угодно, но уже количество функций Мастера — около 200! — говорит о том, что овладение системой на уровне языка потребует времени и усилий. Однако откладывать начало практической работы с системой до завершения изучения языка не придется — входящая в систему программа "Лоцман" позволяет в диалоговом режиме, без использования процедурного программирования выполнить всю работу по созданию базы данных (задать структуру базы, описания множеств и полей, различные варианты индексации, необходимые реляционные связи). В случае затруднений всегда можно обратиться к спасительному пункту "Помощь" в меню и получить контекстно-зависимую подсказку.

Если же программистов в вашей организации нет вовсе, а нужда в сложной специализированной программной системе имеется — лучше обратиться к профессиональным "мастеровым" для разработки заказной прикладной системы.

Поставляется также ряд готовых систем, работающих в среде Мастера и адресованных конечному пользователю — экономисту, инженеру, бухгалтеру, редактору.

Учиться, учиться и учиться, как завещал Великий...

А теперь вернемся в старую коммуналку на Кожевнической, чтобы поговорить о третьем ките "Микроинформа" — его учебном центре. Здесь было, мягко говоря, тесновато. Программно-технический отдел в полном составе помещался в одной разгороженной щитами-ширмами комнате, и чтобы добраться до компьютера, иногда приходилось решать напоминающую шахматы задачу с ходами "конем" между столами и несколькими рокировками с коллегами. Соседняя комната, самая большая, была отведена под компьютерный класс. Класс этот работал с полной нагрузкой с момента рождения "Микроинформа" до того дня, когда ему на смену пришли новые просторные залы.

Что можно сказать сегодня об обучении в "Микроинформе"? Даже неискушенный неофит получает здесь хороший стартовый толчок. Он преодолевает первый психологический барьер в работе с компьютером — ведь "живая" машина стоит на каждом столе и даже в гостинице, где "Микроинформ" размещает иногородних слушателей. После окончания 8-часовых занятий с перерывом на обед в соседней ресторации не возбраняется засиживаться в классе до тех пор, пока не ушел домой последний программист, а этот момент при окончании официального рабочего дня в 18.00 частенько затягивается почти до полуночи.

Навыки уверенной работы с Лексиконом и SuperCalc позволят слушателю, вернувшись на родное предприятие, сразу же порадовать начальство быстрой подготовкой деловых расчетов и документов вместо бесконечной укладки кубиков "Тетриса" или странствий по злодейским подземельям и звездных войн. Получив также кое-какие познания о начале всех начал — операционной системе MS-DOS и о системах управления базами данных и интегрированных средах (на примерах классической dBASEIII+ и Мастера), слушатель сможет в дальнейшем осознанно подойти к выбору программных средств для решения более сложных задач и справиться с их освоением.

Что же до специалистов, опытных пользователей и программистов, для них у руководителя учебного центра О.Ф.Литвина имеется следующее меню:

- системное сопровождение, эксплуатация и ремонт ПЭВМ;
- установка и администрирование ЛВС, программное обеспечение Novell, прикладные системы в сети;
- разработка программ обработки данных на языке Си;
- разработка диалоговых информационных систем в среде СУБД dBASE, Clipper, в инструментальной системе Мастер;
- настольная редакционно-издательская система Ventura Publisher.

Особо следует сказать о сотрудничестве с фирмой Novell: в учебном центре "Микроинформа" гордо реет красный стяг с надписью "Novell Education", осеняющий авторизованный класс фирмы. Здесь готовят дипломированных специалистов по компьютерным сетям точно так же, как в любом классе Novell в любой стране мира.

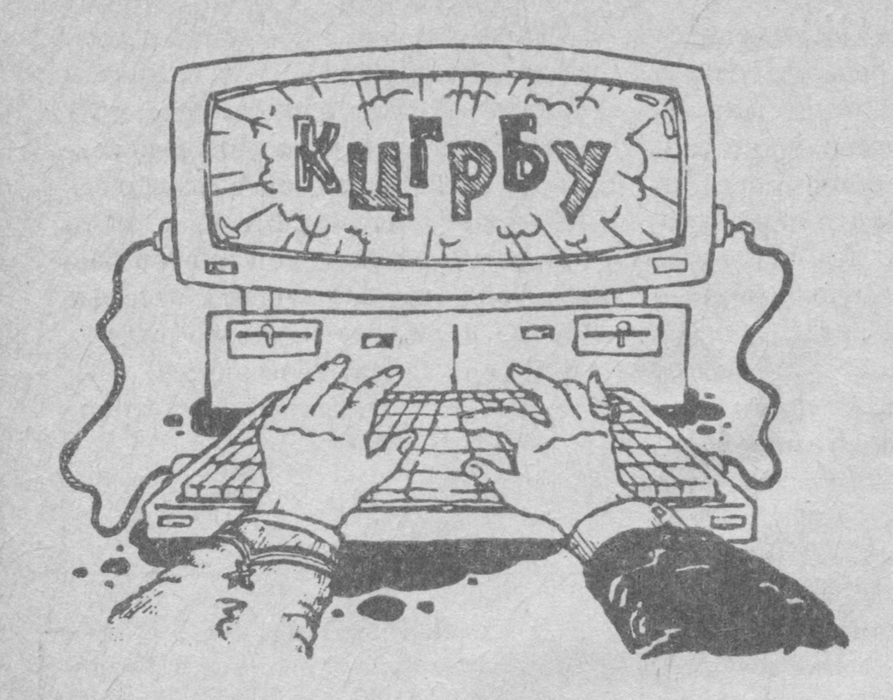
На сладкое предлагается индивидуальное и групповое обучение по конкретным целевым программам. Обучение в "Микроинформе" проходили группы спе-

циалистов министерств энергетики и гражданской авиации, нефтяники и сотрудники московского представительства Rank Xerox.

Что ждет нас впереди?

А впереди ждет нас то же самое, компания не собирается сворачивать с дороги, по которой движется уже пятый год. В дальнейших планах "Микроинформа" есть и расширение сети существующих филиалов, и заключение новых контрактов на поставку предприятиям и хозяйствам страны самых современных персональных компьютеров, есть и новинки типа создания национальных версий продуктов, поддерживающих одновременную работу на трех языках. Уже подготовлены украинская, казахская и грузинская версии Лексикона, идет работа и над другими. Группа системных программистов под руководством Веселова ведет разработку проекта Мастер-2, который, в отличие от прежнего Мастера, должен стать не замкнутой программной системой, а открытой технологией, переносимой на различные оконные платформы. Дел впереди — непочатый край, но главное то, что "Микроинформ" нашел свой путь к компьютерному храму и идет по нему, не сбиваясь на другие тропинки.

О.Дергунова,
Г.Шмерлинг



КГБ и ЦРУ — братья навек?

Компактные лазерные диски только для чтения CD-ROM (Compact Disc Read Only Memory), как недорогое, надежное и удобное средство хранения и распространения информации, стремительно завоевывают все более широкую популярность во всем мире. Первоначально, еще несколько лет назад, CD-ROM рассматривались лишь как переходный этап к технологии уже существующих, но значительно более дорогих оптических дисков WORM (Write Once Read Many), предназначенных для однократной записи данных.

Простая и хорошо освоенная тиражируемость 5,25-дюймовых дисков CD-ROM, технологически аналогичная обычным звуковым компакт-дискам, и весьма низкая их себестоимость — всего лишь около одного-двух долларов на один диск при массовом тираже — обеспечивают огромный интерес к этой многообещающей продукции у многочисленных издательств CD-ROM, которые теперь появляются в огромном количестве, словно грибы после дождя. В Соединенных Штатах уже имеется несколько сотен издательств дисков CD-ROM, организована сеть оптовой продажи и розничной торговли по почте, выпускаются сводные каталоги выпущенных компактных дисков, открываются отделы по продаже CD-ROM в книжных и музыкальных магазинах. Буквально ежедневно появляются все новые и новые наименования дисков CD-ROM, которые уже сегодня можно не только купить, но и получить на прокат во многих американских библиотеках.

Большинство американских издательств помещают на диски CD-ROM всевозможную техническую, медицинскую, коммерческую, юридическую, религиозную, справочную и иную текстовую информацию, проиллюстрированную изображениями в графическом формате PCX. Доступ пользователя к огромному объему текстовой информации (а полная емкость одного диска CD-ROM достигает 640 Мбайт!) облегчается тем, что ВСЕ слова в таких текстах обычно бывают проиндексированы, и поэтому специальная система доступа обеспечивает практически мгновенный поиск любого слова в тексте сколь угодно большого объема. Увы, из-за того, что CD-ROM сегодня наиболее популярны именно в США, в большинстве дисков CD-ROM информация представлена только на английском языке, разумеется, за исключением дисков, содержащих учебные пособия по иностранным языкам и словари.

Большая емкость CD-ROM позволяет размещать на таких дисках даже весьма громоздкие цветные графические файлы с репродукциями картин, фотографий, географических карт и иных изображений. Некоторые издательства добавляют на своих CD-ROM еще и звуковые эффекты — отрывки музыкальных произведений, звуки, издаваемые животными и птицами, и т.п. Считать такие издания настоящим мультимедиа, пожалуй, еще нельзя, так как в большинстве случаев отсутствует тесная органическая взаимосвязь между звуком и объемным живым изображением. Однако и такие опыты несомненно интересны и весьма информативны, особенно там, где компьютеры применяются в качестве справочного или учебного средства — в библиотеках, музеях, информационных агентствах, университетах и колледжах.

В странах Западной Европы, в Юго-Восточной Азии и остальном мире популярность CD-ROM пока еще остается заметно более низкой, чем в США. Но в связи с неуклонным снижением стоимости дисководов CD-ROM примерно на 30% ежегодно, в последнее время повсеместно отмечается стремительный рост интереса к этим новым средствам хранения и распространения информации. Некоторые модели дисководов CD-ROM сегодня уже продаются в розницу всего лишь за 200—250 долл. А

ведь такой дисковод не только позволяет считывать цифровые данные для обработки в персональных компьютерах, но при подключении к акустическому усилителю со стереоколонками обеспечивает высококачественное воспроизведение музыкальных компактных дисков, достигших уже настолько прочного массового распространения за рубежом, что это повлекло почти повсеместное резкое сокращение выпуска обычных грампластинок.

Вместительные и недорогие диски CD-ROM обычно рассматриваются как путь, прямиком ведущий к будущей эпохе мультимедиа. Этим объясняется энергичная поддержка индустрии CD-ROM со стороны таких гигантов, как IBM и Microsoft, делающих ставку на массовое распространение мультимедиа как среды общения пользователей с компьютерами следующих поколений. Многие фирмы уже сегодня выпускают персональные компьютеры, ориентированные именно на это перспективное применение компьютеров. Такие системы обычно поставляются с одним дисководом для гибких дисков и дисководом для CD-ROM, но вовсе без жесткого диска, который, впрочем, может быть установлен за дополнительную плату. Корпорация IBM выпускает семейство своих недорогих компьютеров PS/1, с ценой за минимальную конфигурацию менее 1000 долл., которые с конца 1990 года оснащаются встроенным дисководом CD-ROM.

Интерфейс SCSI, применяемый сегодня в большинстве дисководов CD-ROM, позволяет последовательно подключать в цепочку до семи таких дисководов как к IBM-совместимым компьютерам через дополнительную плату, так и непосредственно к компьютерам Macintosh. Нередко такие компьютеры с дисководами CD-ROM снабжаются дополнительными аудиоплатами, высококачественно воспроизводящими Hi-Fi стереозвук через наушники или колонки. Предполагается, и не без оснований, что нарастающая популярность CD-ROM сможет несколько оживить мировую компьютерную индустрию, подающую в последние годы признаки некоторого застоя из-за общего экономического спада и наступающего перенасыщения рынка.

Известно, что авторские права и права на интеллектуальную собственность весьма высоко оплачиваются во всех цивилизованных странах, если интеллектуальный продукт удовлетворяет реальную общественную потребность и имеет рыночный спрос. Поэтому в настоящее время большинство CD-ROM-дисков в целях снижения рыночной цены и облегчения маркетинга содержат лишь произведения, являющиеся public domain, то есть общественным достоянием, на которые не распространяется copyright и за которые не требуется платить авторские гонорары. Таков, например, полный телефонный справочник Соединенных Штатов PhoneDisc USA Residential на двух дисках CD-ROM, который содержит 90 миллионов имен, адресов и телефонных номеров абонентов. В каталогах дисков CD-ROM можно найти полные собрания сочинений Шекспира, Марка Твена, Конан Дойла или несколько вариантов Библии, представленные на одном диске.

Однако на рынке уже начинают появляться и диски с произведениями ныне живущих авторов литературных бестселлеров. Библиотека Конгресса США недавно объявила о начале работы по переводу всех своих фондов в электронную форму для последующего распространения на дисках CD-ROM. Можно уверенно предположить, что к моменту завершения этого грандиозного проекта, который потребует не одно десятилетие кропотливого труда, большинство книг и журналов во всем мире будут выходить исключительно в электронной форме на какой-то разновидности сегодняшних дисков CD-ROM. Впрочем, уже сегодня появились первые периодические издания, распространяемые только на дисках CD-ROM.

Например, американский электронный журнал Nautilus выпускается в двух версиях для Macintosh и для Windows, а его годовая подписка на 12 выпусков стоит около 150 долл., включая доставку авиапочтой через океан. Пробную бесплатную копию журнала Nautilus на CD-ROM и его годовую подписку можно заказать по адресу:

NAUTILUS
7001 Discovery Blvd.
Dublin, Ohio 43017-8066
U.S.A.

В КомпьютерПресс № 2´92 уже упоминалось американское издательство CD-ROM-дисков Quanta Press. Это издательство известно своим лазерным CD-ROM-диском CIA World Factbook, который числится в десятке бестселлеров CD-ROM в США наряду с такими известными изданиями, как Microsoft Bookshelf, Guinness Disc of Records, Compton´s MultiMedia Encyclopedia и Grolier´s Electronic Encyclopedia. Издательство Quanta Press только что объявило о выпуске своего нового сенсационного диска, который обещает принести издательству не меньший успех. Этот диск называется KGB World Factbook. В пресс-релизе по поводу выпуска этого диска издательство утверждает, что хотя коммунизм и СССР покинули историческую сцену, КГБ благополучно преодолел все опасности и неплохо чувствует себя в России Бориса Ельцина.

KGB World Factbook структурно схож с диском CIA World Factbook. Здесь содержится весьма подробная несекретная информация о 249 странах и территориях всего мира. Справочный текст на английском языке включает важнейшую информацию об этих странах и территориях: географические, климатические, статистические, экономические, демографические сведения, данные о развитии инфраструктуры, о властях, об административном и правительственном устройстве, о внешней торговле, о важнейших политических событиях, партиях и

Итак, вечные противники — КГБ и ЦРУ, которых было водой не разлить, опять сошлись вместе на дисках CD-ROM, дополняя и уточняя друг друга. Желающие заказать эти диски могут обращаться к директору маркетинга издательства Dennis E. Burke по адресу:

Quanta Press, Inc.
1313 Fifth Street SE, Suite 208C
Minneapolis, MN 55414
U.S.A.
Телефон: (612)379-3956
Телефакс: (612)623-4570

лидерах, о дипломатических представительствах, о транспорте и связи, о флоте и авиации, о вооруженных силах. Кроме того, на диске можно найти справочник по переводу различных мер и весов, о поясном времени, о структуре и институтах Организации Объединенных Наций. На диске имеются интерактивные черно-белые карты всего мира, а также руководство по работе с этим справочником, глоссарий и список принятых сокращений. И хотя в пресс-релизе Quanta Press подчеркивается, что диск KGB World Factbook содержит только те сведения, которые доступны всем из открытых источников, то есть являются public domain, можно с уверенностью утверждать, что ни в одной энциклопедии или справочнике столь актуальных, обширных и разнообразных сведений, собранных воедино, найти невозможно.

Как и большинство дисков CD-ROM, диск KGB World Factbook выпускается в двух форматах — для DOS и для Mac. Рекомендованная розничная цена этого диска составляет 99 долл. плюс 15 долл. за доставку авиапочтой. Дилерам и оптовым покупателям предоставляются весьма существенные скидки. Quanta Press предлагает с большой скидкой вместе оба диска KGB World Factbook и CIA World Factbook. Рекомендованная розничная цена за оба диска 149 долл. плюс 15 долл. — доставка через океан.

Пожалуй, диск CD-ROM с именем КГБ в наименовании впервые появляется на мировом рынке. Издательство Quanta Press готово рассмотреть предложения других организаций и частных лиц на издание имеющейся у них информации, разумеется, если такая информация может представлять интерес для тиражирования в электронной форме. Причем это может быть не только текстовая, но и графическая информация. Важно только, чтобы в случае таких предложений не было почвы для сомнений о правах на предлагаемую интеллектуальную собственность.

А.Петроченков

НОВОСТИ

Corel Systems Corporation анонсирует новую версию CorelDRAW! 3.0 — самого полного графического пакета

Оттава, Онтарио, Канада — 15 мая 1992 года фирма Corel объявила о выпуске CorelDRAW 3.0. Снабженная тремя новыми мощными модулями и улучшенными версиями старых, новая версия CorelDRAW является самым полным на сегодня графическим пакетом из имеющихся в мире.

CorelDraw 3.0 сочетает в себе мощные программы векторной графики, деловой графики, обработки цветных изображений и презентационную программу. CorelDraw 3.0 будет продаваться по цене 595 американских долларов или 695 канадских долларов с июня этого года.

Среди новых возможностей CorelDraw 3.0:

- предварительный просмотр созданных графических образов;
- неограниченное количество слоев, которые могут быть соединены, спрятаны, напечатаны, раскрашены;
- использование направляющих объекта, которые могут быть раскрашены, показаны, спрятаны;
- редактирование текста на экране, проверка орфографии, тезаурус, перенос слов;
- простой доступ к наиболее часто используемым режимам вследствие введения прокручивающихся окон;
- коммуникативность — CorelDraw 3.0 действует как клиент и сервер OLE.

Пакет включает вновь введенные или улучшенные эффекты, позволяющие удобно размещать текст вдоль траектории, полностью управлять формой, цветом, освещением и расположением объекта в трехмерном пространстве с помощью штамповки, применять раскраску к группам объектов и к объектам вдоль траектории. Теперь траекторию можно изменять интерактивно. CorelDraw 3.0 включает также возможность градиентной раскраски.

Кроме того, CorelDraw 3.0 включает новые и переработанные модули:

- CorelCHART. Более 100 различных видов графиков от пиктограмм до трехмерных графиков, построение трехмерных диаграмм;
- CorelPHOTO-PAINT. Объединяет в удобный для использования модуль редактирования 24-битных образов мощные средства фоторетуши и возможности раскрашивания;
- CorelSHOW. Позволяет извлекать элементы из различных приложений, включая CorelDraw, CorelCHART, CorelPHOTO-PAINT и другие OLE-серверы, в единые многостраничные изображения и слайд-шоу. Возможен доступ к созданным профессионалами библиотекам заготовок кадров и роликов Autodesk Animator´a;
- CorelMOSAIC. Просматривает, управляет, производит поиск, пакетную печать, импортирование и экспортирование файлов форматов CDR, BMP, Windows DIP, EPS, GIF, PCX и TIFF;
- CorelTRACE. Преобразует черно-белое растровое изображение в гладкую векторную графику.

CorelDraw 3.0 также включает 153 шрифта True Type, более 4600 профессионально нарисованных символьных и кадровых изображений, доступ к стандартным промышленным системам задания цвета при печати. Вместе с пакетом CorelDraw 3.0 бесплатно поставляется CD-ROM с 14,000 графических образов CorelDRAW, более чем сотней TrueType-шрифтов, полной библиотекой шрифтов Corel в формате PostScript Type 1 и подробными вводными курсами.

Модули, входящие в состав CorelDraw 3.0, заменяют программное обеспечение, которое можно приобрести у различных поставщиков примерно за 2000 долл.

Компания объявила о выпуске в этом году версии CorelDraw 3.0 для Macintosh и для сетей. Также разрабатывается версия для UNIX.

Microsoft объявляет о грандиозных планах

Директор по развитию бизнеса в бывшем СССР фирмы Microsoft Ричард Хеврон (Richard Hevron) дал пресс-конференцию в Москве, где объявил о далеко идущих планах фирмы в нашей стране.

"Мы приняли решение в течение ближайших нескольких месяцев открыть офис фирмы в Москве. Офис будет, как обычно, на сто процентов принадлежать нашей фирме, а работать в нем будут в основном здешние люди в количестве 10-22 человек".

Какую сумму фирма планирует вложить в развитие своей московской инфраструктуры, директор сказать отказался.

Фирма в течение следующих 6-12 месяцев начнет продавать свои основные продукты за рубли по приемлемым ценам. "Мы начнем с самых популярных продуктов, а потом будем расширять объем наших предложений",— сказал г-н Хеврон.

Компания Microsoft также подписала договор о деловом сотрудничестве с фирмой РосПрограммИмпорт (РПИ), которая уже долгое время представляет интересы Microsoft в СНГ. Частная фирма РПИ проводит работы по локализации Windows 3.1 для российского рынка. Этот продукт должен появиться в продаже в ноябре этого года. Две другие программы — WinWorks 1.0 и Excel for Windows — в первом квартале следующего.

Последним заявлением Microsoft было то, что фирма до конца этого года подпишет договоры с книгоиздателями и напечатает ряд своих толстых и интересных книжек для программистов и пользователей на русском языке.

Некоторое время назад главный бухгалтер фирмы Microsoft, говоря об очередных квартальных результатах работы компании, объявил, что в течение этого и следующего годов фирма откроет свои торговые представительства в бывшем СССР и Восточной Европе. Похоже, эти планы уже начинают сбываться.

CHIPS подписывает контракт с ComputerLand. Summit Systems получает те же компьютеры, что и ComputerLand в США

22 июня 1992 года один из лидеров электронной промышленности США — фирма CHIPS — подписала контракт с компанией ComputerLand, предоставившей ей свою сеть розничной торговли компьютерами для распространения новейших персональных компьютеров ModTech I. Особенностью этой новинки компьютерного рынка является необычная архитектура компьютера: процессор/сопроцессор и кэш-память расположены на сменном процессорном модуле, что облегчает замену устаревшего процессора на новый (например, 486 на 586 или ?86).

Процессорные модули в 5-10 раз дешевле самого компьютера, таким образом, модернизация парка вычислительной техники для предприятий или отдельных пользователей будет обходиться во столько же раз дешевле.

На презентации новых компьютеров в Новгороде во время встречи директоров СП "Диалог" с представителем компании CHIPS Джефом Грамером было зарегистрировано рекордное время (11 секунд!) смены процессорного модуля.

СП Саммит Системс (официальный представитель фирмы CHIPS в России и Беларуси), в свою очередь, начинает поставки этих компьютеров на наш рынок. Сегодня эти компьютеры поддерживают процессорные блоки 386, 486SX и 486DX с тактовыми частотами до 50 МГц.

Символично также и название семейства этих компьютеров — Хамелеон, так как, по словам менеджера по маркетингу Дагласа МакКаллама, эти компьютеры могут также быстро менять свой процессор, как одноименное животное — свой цвет.

21 июня 1992 года в Пало Альто (Калифорния) прошла презентация новых сетевых адаптеров Ethernet фирмы Hewlett-Packard. Адаптеры серии HP PC LAN Adapter/16TP Plus предназначены

для IBM-совместимых персональных компьютеров с шиной ISA, объединяемых в локальную сеть.

Экспертиза, проведенная независимой тестовой лабораторией NSTL (National Software Testing Laboratories), показала, что по критериям общей производительности и отношению стоимость/производительность новые сетевые адаптеры Hewlett-Packard превосходят аналогичные изделия таких фирм, как 3COM, Intel, Novell, SMC. В частности, новые адаптеры меньше загружают центральный процессор компьютера. Это особенно важно при работе в многозадачном режиме, например, в среде UNIX или MS Windows.

Адаптеры поставляются вместе с программными драйверами для таких сетевых операционных систем, как Novell NetWare 286 и 386, Microsoft LAN Manager 2.1 и Banyan Vines 4.1. Для использования в составе рабочих станций, не оснащенных дисковыми устройствами, платы адаптеров снабжаются специальными загрузочными ПЗУ (Boot-ROM). Использование заказной СБИС сетевого контроллера (ASIC) позволяет сократить количество микросхем на плате адаптера примерно на 20 штук. Это обстоятельство благотворно влияет не только на производительность, но и на надежность новых изделий — гарантийное обслуживание новых адаптеров обеспечивается в течение всего их срока работы.

Фирмой Toshiba разработан 2,5-дюймовый жесткий диск объемом 200 Мбайт. До настоящего времени устройство объемом 180 Мбайт было наибольшим среди выпускаемых серийно.

2,5-дюймовый жесткий диск размером 70x100x19 мм создан на базе диска MK-2124FC емкостью 130 Мбайт и полностью совместим с ним.

Toshiba уже производит большое количество жестких дисков объемом 200 Мбайт и теперь готова запустить новую производственную линию по массовому выпуску новых малогабаритных устройств на заводе в Токио. Там же Toshiba производит устройства объемом 130 и 80 Мбайт.

Фирмой уже разрабатывается следующая версия 2,5-дюймового жесткого диска объемом 200 Мбайт.

Newsbytes, February 11, 1992

Поставьте будущее себе на стол!



- Ваши потребности растут? Наш компьютер совершенствуется!
- Надежность под знаком Intel, CHIPS, Quantum, Sony
- Уникальная производительность - убедитесь сами
- 2 года гарантии обеспечены быстрым обслуживанием в 50-ти городах от Бреста до Находки
- Комплексный подход: 286-е компьютеры, принтеры, сетевое оборудование и программное обеспечение

Модульный Саммит

386-33/ 486sx-25/ 486-40MHz
цветной Super VGA монитор (1024x768)
ОЗУ 4 Мб (макс. 32 Мб)
Жесткий диск 105/210 Мб (17/15 мс)
Русифицированная клавиатура
Лицензионная ДОС
2 года гарантии

МОСКВА (095) 299 1162
ВНЕ МОСКВЫ (0172) 973 119

34-27 72



Задумано в Америке. Сделано для России.